

Administration Guide

vCenter Orchestrator 4.0

EN-000127-00



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

<http://www.vmware.com/support/>

The VMware Web site also provides the latest product updates.

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

docfeedback@vmware.com

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

VMware, the VMware "boxes" logo and design, Virtual SMP, and VMotion are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

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

Contents

About This Book	5
1 Introduction to VMware vCenter Orchestrator	7
Key Features of the Orchestrator Platform	7
Orchestrator User Roles and Related Tasks	8
Orchestrator Architecture	9
2 The Orchestrator Client Workspace	11
Log in to the Orchestrator Client	11
3 Managing Workflows	13
The Workflows View	13
Components of the Workflows View	14
Standard Workflows in the Workflow Library	14
Key Concepts of Workflows	15
Workflow Attributes	15
Workflow Parameters	15
Workflow Schema	16
View Workflow Schema	16
Workflow Tokens	16
Workflow Token States	17
Set User Permissions on a Workflow	18
Run a Workflow	19
4 Managing Actions	21
The Actions View	21
Create an Action	22
Duplicate an Action	22
Export an Action	23
Import an Action	23
Move an Action	23
Find Elements That Implement an Action	24
5 Managing Tasks	25
The Tasks View	25
Create a Task	25
Edit the Task Recurrence Pattern	26
6 Policies	27

7	Using Packages	29
	The Packages View	29
	Create a Package	30
	Set User Permissions on a Package	31
	Export a Package	31
	Import a Package	32
	Get and Synchronize a Remote Package	33
8	Default Orchestrator Plug-Ins	35
9	Modify the Number of Objects a Plug-In Search Obtains	39
10	Access Java Classes in JavaScript	41
11	Maintenance and Recovery	43
	Change the Size of Server Logs	43
	Index	45

About This Book

The *VMware vCenter Orchestrator 4.0 Administration Guide* provides information and instructions about using and maintaining VMware® vCenter Orchestrator. It also describes how to manage workflows, plug-ins, packages, inventory and policies.

Intended Audience

This book is intended for advanced vCenter administrators and experienced system administrators who are familiar with virtual machine technology and datacenter operations, as well as anyone who wants to:

- Automate frequently repeated processes related to the management of the virtual environment.
- Manage multiple automated processes across and among heterogeneous systems.
- Provide transparency in IT processes by centralizing automated scripts.
- React faster to unplanned changes in the virtual environment.

Document Feedback

VMware welcomes your suggestions for improving our documentation. If you have comments, send your feedback to docfeedback@vmware.com.

Technical Support and Education Resources

The following technical support resources are available to you. To access the current version of this book and other books, go to <http://www.vmware.com/support/pubs>.

Online and Telephone Support

To use online support to submit technical support requests, view your product and contract information, and register your products, go to <http://www.vmware.com/support>.

Customers with appropriate support contracts should use telephone support for the fastest response on priority 1 issues. Go to http://www.vmware.com/support/phone_support.html.

Support Offerings

To find out how VMware support offerings can help meet your business needs, go to <http://www.vmware.com/support/services>.

VMware Professional Services

VMware Education Services courses offer extensive hands-on labs, case study examples, and course materials designed to be used as on-the-job reference tools. Courses are available onsite, in the classroom, and live online. For onsite pilot programs and implementation best practices, VMware Consulting

Services provides offerings to help you assess, plan, build, and manage your virtual environment. To access information about education classes, certification programs, and consulting services, go to <http://www.vmware.com/services>.

Introduction to VMware vCenter Orchestrator

1

VMware vCenter Orchestrator is a development and process-automation platform that provides a library of extensible workflows to allow you to create and run automated, configurable processes to manage the VMware vCenter infrastructure.

Orchestrator exposes every operation in the vCenter Server API, allowing you to integrate all of these operations into your automated processes. Orchestrator also allows you to integrate with other management and administration solutions through its open plug-in architecture.

This chapter includes the following topics:

- [“Key Features of the Orchestrator Platform,”](#) on page 7
- [“Orchestrator User Roles and Related Tasks,”](#) on page 8
- [“Orchestrator Architecture,”](#) on page 9

Key Features of the Orchestrator Platform

Orchestrator is composed of three distinct layers: an orchestration platform that provides the common features required for an orchestration tool, a plug-in architecture to integrate control of subsystems, and a library of preexisting processes. Orchestrator is an open platform that can be extended with new plug-ins and libraries, and can be integrated into larger SOAP architectures through a set of APIs.

The following list presents the key Orchestrator features.

Persistence	Production grade external databases are used to store relevant information, such as processes, states, and configuration information.
Central management	Orchestrator provides a central way to manage your processes. The application server-based platform, with full version history, allows you to have scripts and process-related primitives in one place. This way, you can avoid scripts without versioning and proper change control spread on your servers.
Check-pointing	Every step of a process is saved in the database, which allows you to restart the server without losing state and context. This feature is especially useful for long-running processes.
Versioning	All Orchestrator Platform objects have an associated version history. This feature allows basic change management when distributing processes to different project stages or locations.

Scripting engine	<p>The Mozilla Rhino JavaScript engine provides a way to create new building blocks for Orchestrator Platform. The scripting engine is enhanced with basic version control, variable type checking, name space management and exception handling. It can be used in the following building blocks:</p> <ul style="list-style-type: none"> ■ Actions ■ Workflows ■ Policies
Workflow engine	<p>The workflow engine allows you to capture business processes. It uses one of the following methods to create a step-by-step automation:</p> <ul style="list-style-type: none"> ■ Building blocks of the library ■ Building blocks provided by the customer ■ Plug-ins <p>Users, a schedule, or a policy can start workflows.</p>
Policy engine	<p>The policy engine allows monitoring and event generation to react to changing conditions. Policies can aggregate events from the platform or any of the plug-ins, which allows you to handle changing conditions on any of the integrated technologies.</p>
Web 2.0 front end	<p>The Web 2.0 front end allows new possibilities of expression and flexibility. It provides a library of user customizable components to access vCO orchestrated objects and uses Ajax technology to dynamically update content without reloading complete pages.</p>
Security	<p>Orchestrator provides the following advanced security functions:</p> <ul style="list-style-type: none"> ■ Public Key Infrastructure (PKI) to sign and encrypt content imported and exported between servers ■ Digital Rights Management (DRM) to control how exported content might be viewed, edited and redistributed ■ Secure Sockets Layer (SSL) encrypted communications between the desktop client and the server and HTTPS access to the Web front end. ■ Advanced access rights management to provide control over access to processes and the objects manipulated by these processes.

Orchestrator User Roles and Related Tasks

vCenter Orchestrator provides different tools and interfaces based on the specific responsibilities of the three global user roles: Administrators, Developers, and End Users.

Administrators	<p>This role has full access to all of the Orchestrator platform capabilities. Basic administrative tasks include the following items:</p> <ul style="list-style-type: none"> ■ Installing and configuring Orchestrator ■ Managing access rights for Orchestrator and applications ■ Importing and exporting packages ■ Enabling and disabling Web views
-----------------------	--

- Running workflows and scheduling tasks
- Managing version control of imported elements

Developers

Users in this role are granted access to the Orchestrator client interface and have the following responsibilities:

- Creating applications to extend the Orchestrator platform functionality
- Automating processes by customizing existing workflows and creating new workflows
- Customizing Web front ends for these processes, using Web 2.0 technologies

End Users

Users in this role are granted access to only the Web front end. They can run and schedule workflows and policies.

Orchestrator Architecture

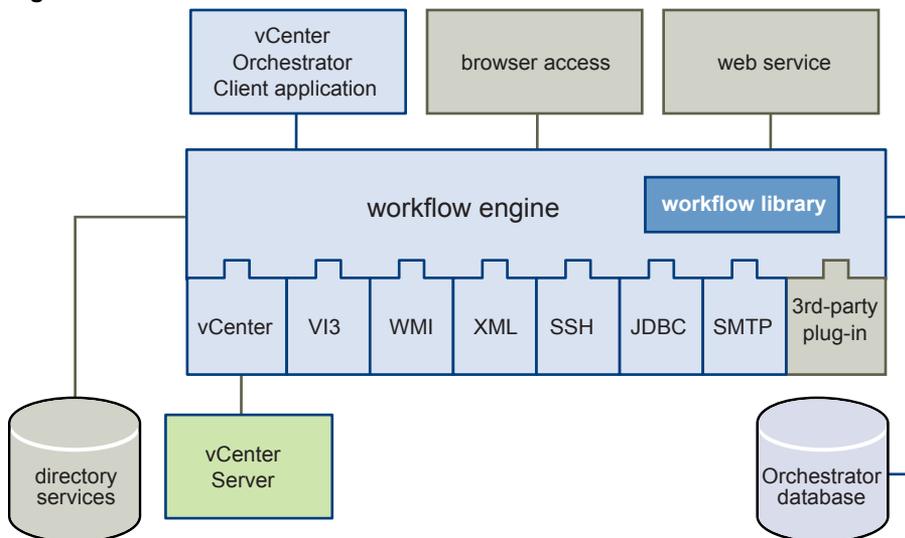
Orchestrator contains a workflow library and workflow engine to allow you to create and run workflows that automate orchestration processes. You run workflows on the objects of different technologies that Orchestrator accesses through a series of plug-ins.

Orchestrator provides a standard set of plug-ins, including a plug-in to VMware vCenter Server 4.0, to allow you to orchestrate tasks in the different environments that the plug-ins expose.

Orchestrator also presents an open architecture to allow you to plug in external third-party applications to the orchestration platform. You can run workflows on the objects of the plugged-in technologies that you define yourself. Orchestrator connects to a directory services server to manage user accounts, and to a database to store information from the workflows that it runs. You can access Orchestrator and the workflows and objects it exposes through the Orchestrator client interface, through a Web browser, or through Web services.

Figure 1-1 shows the architecture of Orchestrator.

Figure 1-1. VMware vCenter Orchestrator Architecture



The Orchestrator Client Workspace

The Orchestrator client is an easy-to-use desktop application that allows you to perform daily administration tasks such as importing packages, running and scheduling workflows and policies, and managing user permissions. The Orchestrator client also serves as an IDE for creating or customizing workflows.

Log in to the Orchestrator Client

To perform general administration tasks, you must log in to the Orchestrator client interface.

Prerequisites

All components of the Orchestrator server must be configured and the VMware vCenter Orchestrator Server service must be running.

Procedure

- 1 Access the Orchestrator Client interface.

On Windows, select **Start > Programs > VMware > vCenter Orchestrator Client**.

- 2 Log in using the Orchestrator user name and password.

The **My Orchestrator** view appears. This view summarizes the recent activities on the server, shows pending and running workflows, running policies, scheduled tasks, completed workflows, and elements you recently edited.

What to do next

You can import a package, start a workflow, or set root access rights on the system.

Managing Workflows

A workflow is a succession of actions and decisions that are run sequentially until they arrive at a specific result. Orchestrator provides a library of workflows that perform common management tasks according to best practices. 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, complete 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 information 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 Orchestrator client interface's integrated development environment (IDE), that provides access to the workflow engine. The workflow engine assembles workflows from the Orchestrator's libraries of predefined objects and actions. The workflow engine can also take objects from external libraries that you plug in to Orchestrator. This ability allows you to customize processes or implement functions that third-party applications provide.

This chapter includes the following topics:

- [“The Workflows View,”](#) on page 13
- [“Standard Workflows in the Workflow Library,”](#) on page 14
- [“Key Concepts of Workflows,”](#) on page 15
- [“Set User Permissions on a Workflow,”](#) on page 18
- [“Run a Workflow,”](#) on page 19

The Workflows View

The Orchestrator client interface features a **Workflows** view that provides access to the Orchestrator libraries of workflows.

The **Workflows** view allows you to view information about each workflow, create, edit, run workflows, and interact with the workflows.



The Orchestrator client uses the following icon to identify workflows:

Components of the Workflows View

The **Workflows** view consists of a set of tabs that show different types of information about a particular workflow.

General	Displays general information about the workflow, including its name, its version number, the permissions, a description, and a list of the workflow's global attributes.
Inputs	Lists all the input parameters that the workflow needs when it runs.
Outputs	Lists the parameters that the workflow returns when it runs.
Schema	Shows a graphical representation of the workflow. Clicking an element in the schema shows information about that element in the bottom half of the Workflows view.
Presentation	Constructs the input parameters dialog box that users see when they run a workflow. You define the groups in which the input parameters appear in the dialog box and provide descriptions to help users provide the correct parameters. You also define any parameter properties or constraints.
Parameters Reference	Shows all the input and output parameters in a single view, with the descriptions and constraints that you define in the Presentation tab. The tab also identifies the schema element that consumes or generates a parameter. You can optionally view the workflow attributes in this tab by clicking Show Attributes .
Executions	Provides details about the different runs of a particular workflow. This information includes the workflow's running status, the user who started it, and the time and date when the workflow started and ended.
Events	Provides information about each event that occurs while the workflow is running. This information includes the event's running status, the user who started it, and the time and date when the event started and ended. The information is stored in the VMO_LogEvent table in the Orchestrator database.
Permissions	Lists the permissions accorded to users or groups of users to interact with the workflow. The possible permissions are View, Execute, Inspect, Edit, and Admin.

Standard Workflows in the Workflow Library

Orchestrator provides a library of standard workflows that allow you to perform many common actions related to vCenter Server management. You can either use these workflows without changing them, or you can extend and customize them to meet your own purposes. The contents of the workflow library is accessible through the **Workflows** tab in the Orchestrator client.

The workflow library that Orchestrator provides contains standard workflows in the following categories.

JDBC	Test the communication between a workflow and a database by using the JDBC (Java Database Connectivity) plug-in shipped with Orchestrator.
Locking	Demonstrates the locking mechanism for automated processes, that allows workflows to lock the resources they use.
Mail	Send and receive emails from workflows.
Orchestrator	Automate certain common Orchestrator operations.

Refactoring	Refactor workflows and applications that you created to run with VMware Infrastructure 3.5, so that they run with vCenter Server 4.0.
SSH	Implement the Secure Shell v2 (SSH-2) protocol. These workflows allow you to implement remote command and file transfer sessions with password and public key-based authentication. The SSH configuration allows you to specify paths to objects to expose in the Orchestrator Inventory.
vCenter	Access the functions of the vCenter Server 4.0 Server API, so that you can incorporate all of the vCenter Server functions into the management processes that you automate by using Orchestrator.
XML	A Document Object Model (DOM) XML parser that you can implement in automated processes.

Key Concepts of Workflows

Workflows consist of actions, attributes, parameters, and schemas. Orchestrator saves a workflow token every time a workflow runs, recording the details of that specific run of the workflow.

Workflow Attributes

Workflow attributes act as global constants and global variables throughout a workflow. Workflow elements process data that they receive as input parameters, and set the resulting output as workflow attributes.

Read-only workflow attributes act as global constants for a workflow. Writeable attributes act as a workflow's global variables.

A workflow attribute has the following properties:

- Read-only flag
- Name
- Type
- Value
- Linking
- Description

You use attributes to transfer variables between workflow elements. You can obtain attributes in the following ways:

- Define attributes when you create a workflow
- Set the output parameter of a workflow element as a workflow attribute
- Inherit attributes from a configuration element

Workflow Parameters

Workflows receive input parameters and generate output parameters when they run.

Input Parameters

An input parameter is a runtime argument that you, an application, or another workflow or action passes to a workflow or action for it to process when it starts, or while it is running.

Input parameters have the following properties:

- name
- type
- description

After you pass a value for an input parameter to a workflow, you cannot change the parameter's name, type, or description.

Output Parameters

A workflow or action's output parameters represent the result of running that workflow or action. Output parameters can change when a workflow or workflow element runs. While they run, workflows and actions can receive as their input parameters the output parameters of other workflows and actions.

Workflow Schema

A workflow schema is a graphical representation of a workflow that shows the workflow as a flow diagram of interconnected workflow elements.

View Workflow Schema

You view a workflow schema in the schema tab for that workflow in the Orchestrator client.

For information about schema elements and creating and editing workflow schema, see *vCenter Orchestrator Developer's Guide*.

Prerequisites

You must be granted the **Inspect** privilege to view schema and scripting.

Procedure

- 1 Click the **Workflows** view in the Orchestrator client.
- 2 Navigate to a workflow in the workflow hierarchical list.
- 3 Click the workflow.

Information about that workflow appears in the right pane.

- 4 Select the **Schema** tab in the right pane.

You see the graphical representation of the workflow.

What to do next

You can drag schema elements from the workflow palette on the left into the workflow schema diagram to edit the workflow schema.

Workflow Tokens

A workflow token represents a workflow that is running or has run.

A workflow is an abstract description of a process that defines a generic sequence of steps and a generic set of required input parameters. When you run a workflow with a set of real input parameters, you receive an instance of this abstract workflow that behaves according to the specific input parameters you give it. This specific instance of a run or running workflow is called a workflow token.

Workflow Token Attributes

Workflow token attributes are the specific parameters with which a workflow token runs. The workflow token attributes are an amalgam of the workflow's global attributes and the specific input and output parameters with which you run the workflow token.

Workflow Token States

Each time you run a workflow, a workflow token appears under that workflow as a new leaf node in the workflows hierarchical list. Clicking a workflow token in the hierarchical list shows tabs in the right pane that show information about the workflow token.

The information shown includes the schema diagram for that workflow, a list of events, the list of the input parameters with which the workflow token ran, and a log file of the workflow. If you click on a workflow token while it is running, you can see the information in the tabs updating in real time.

[Table 3-1](#) describes the workflow token states.

Table 3-1. Workflow Token States

State	Icon	Description
Running		The workflow token is running.
Waiting for User Interaction		The workflow token is suspended while it waits for input parameters from a user interaction, from an external application, or from another workflow.
Waiting for Event or Timer		The workflow token is suspended while it waits for a signal from an external trigger or a timer before resuming. Long-running workflows enter this state while they wait for the signal to resume running. During the waiting period, the workflow threads become passive.
Canceled		The workflow token is canceled by the user, by an external application, or by another workflow.
Failed		The workflow token failed.
Completed		The workflow token ran successfully. However, a completed workflow token might have encountered errors when it ran, if error-handling is part of the workflow definition.

Set User Permissions on a Workflow

You set different levels of permission on a workflow to limit the access that different users or user groups can have to that workflow.

You select the different users and user groups for which to set permissions from the users and user groups in the Orchestrator LDAP server. Orchestrator defines levels of permissions that you can apply to users or groups.

View	The user can view the elements in the workflow, but cannot view the schemas or scripting.
Inspect	The user can view the elements in the workflow, including the schemas and scripting.
Execute	The user can run the workflow
Edit	The user can edit the workflow.
Admin	The user can set permissions on the workflow.

IMPORTANT In production environments, you should limit the permission to edit workflows to Orchestrator administrators only. Allowing other users to edit workflows in production environments can lead to unwanted changes in workflow semantics.

Prerequisites

You must have created a workflow, opened it for editing in the workflow workbench, and added to it the necessary elements.

Procedure

- 1 Click the **Permissions** tab in the workflow workbench.
- 2 Click the **Add access rights** link to define permissions for a new user or user group.
- 3 Search for a user or user group by entering text in the **Search** text box.

The search results show all the users and user groups from the Orchestrator LDAP server that match the search.

- 4 Select a user or user group and click **OK**.
- 5 Right-click the user and select **Add access rights**.
- 6 Check the appropriate check boxes to set the level of permissions for this user and click **OK**.

The levels of permissions are not cumulative. To grant a user permission to view the workflow, inspect the schema and scripting, run and edit the workflow, and change the permissions, you must check all check boxes.

- 7 Click **Save and Close** to exit the package editor.

You set the appropriate user permissions on a workflow.

Run a Workflow

You run a standard Orchestrator workflow to automate a frequently repeated process related to the management of your virtual environment, for example creating a virtual machine in a vCenter Server connected to Orchestrator.

Prerequisites

You must have configured the vCenter plug-in. For details, see *vCenter Orchestrator Installation and Configuration Guide*.

Procedure

- 1 Click the **Workflows** view in the Orchestrator client.
- 2 Select **Library > vCenter > Virtual Machine Management > Basic** to navigate to the **Create VM (Simple)** workflow.
- 3 Right-click the **Create VM (Simple)** workflow and select **Execute Workflow**.

The input parameters dialog box opens.

- 4 Enter the following information to create a virtual machine.

Option	Action
VM name	Enter a name for the virtual machine. For example, orchestrator-test
VM folder	<ol style="list-style-type: none"> a Click the Not Set link. b Press Enter without entering any text in the Search text box. The selection box lists all of the objects that the infrastructure contains of the type VC:VmFolder. c Select a VC:VmFolder object and click Select.
Size of the new disk in GB	Enter an appropriate numeric value.
Memory size in MB	Enter an appropriate numeric value.
Number of virtual CPUs	Select an appropriate number of CPUs from the Number of virtual CPUs drop-down menu.
VM guest OS	Click the Not Set link and select a guest operating system from the list.
Host on which VM will be created	Click the Not Set link and select a host from the proposed list.
Resource pool	<ol style="list-style-type: none"> a Click the Not Set link. b Navigate through the vCenter Server infrastructure hierarchy to the resource pool. c Click Select.
Network to connect to	Click the Not Set link and select a VC:Network object from the list.
Datastore on which the VM will be created	Click the Not Set link and select a VC:Datastore object from the list.

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

A workflow token appears in a leaf node under the **Create VM (Simple)** workflow. The token shows the workflow running icon.

- 6 Click the workflow token to view the status of the workflow as it runs.
- 7 Click the **Events** tab in the workflow token view to follow the progress of the workflow token until it completes.

- 8 Click the **Inventory** view in the Orchestrator client.
- 9 Navigate through the vCenter Server infrastructure hierarchy to the resource pool you defined in [Step 4](#).
The orchestrator-test virtual machine is present in the resource pool.

The **Create VM (Simple)** workflow ran successfully.

What to do next

You can log in vSphere Client and manage the new virtual machine.

Managing Actions

Actions represent individual functions that you use as building blocks in workflows, Web views, and scripts. Actions are JavaScript functions that take multiple input parameters and have a single return value. Actions can call on any object in the Orchestrator API, or on objects in any API that you import into Orchestrator by using a plug-in.

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

When you define actions independently from the workflows that call upon them, you can update or optimize the actions more easily. Defining individual actions also allows other workflows to reuse actions.

This chapter includes the following topics:

- [“The Actions View,”](#) on page 21
- [“Create an Action,”](#) on page 22
- [“Duplicate an Action,”](#) on page 22
- [“Export an Action,”](#) on page 23
- [“Import an Action,”](#) on page 23
- [“Move an Action,”](#) on page 23
- [“Find Elements That Implement an Action,”](#) on page 24

The Actions View

The **Actions** view in the Orchestrator client interface allows you to access the libraries of predefined actions. In the **Actions** view, you can duplicate actions, export them to other workflows or packages, or move them to a different category in the actions hierarchical list.

By expanding the nodes of the actions hierarchical list, you can browse available actions. When you select an action in the list, the right pane displays details about that action.

The **Actions** view presents the following tabs.

General	Displays general information about the action, including its name, its version number, the operations the user is allowed to perform, and a description.
Scripting	Displays the action's return types, input parameters, and the JavaScript code that defines the action's function.
Events	Displays all the events that this action encountered or triggered.
Permissions	Displays which users and user groups have permission to access the action.

Create an Action

When you define an individual function as an action, instead of coding it directly into a scriptable task workflow element, you can expose it in the library for other workflows to use.

Procedure

- 1 Click the **Actions** view in the Orchestrator client.
- 2 Expand the root of the actions hierarchical list and navigate to the module in which you want to create the action.
- 3 Right-click the module and select **Add action**.
- 4 Enter the name of the action in the text box and click **OK**.
- 5 Right-click the action and select **Edit**.
- 6 Click the **Scripting** tab.
- 7 To change the default return type, click the **void** link.
- 8 Add the action input parameters by clicking the arrow icon.
- 9 Write the action script.
- 10 Click **Save and close**.

Your custom action is added to the library of actions.

What to do next

You can use the new custom action in a workflow.

Duplicate an Action

You can duplicate any predefined actions and reuse them in your scripts.

Procedure

- 1 Click the **Actions** view in the Orchestrator client.
- 2 Expand the root of the actions hierarchical list and navigate to the action to duplicate.
- 3 Right-click the action and select **Duplicate action**.
- 4 Enter a name for the new action.
A number is appended to the name of the action if you do not enter a value in this text box.
- 5 For the value of **Action module**, select the module to which you want to add the new action.
- 6 (Optional) Select **No** if you do not want version history to be copied.

When you import an action, its version is compared to the version of the local content, allowing the administrator to decide whether to import it or not.

- 7 Select **Duplicate**.

The new action is available in the library of actions.

What to do next

You can use the action in a workflow.

Export an Action

You can export an action to reuse it in other workflows, policies, or Web views.

Procedure

- 1 Click the **Actions** view in the Orchestrator client.
- 2 Expand the root of the actions hierarchical list and navigate to the action to export.
- 3 Right-click the action and select **Export action**.
- 4 (Optional) Select the **Encrypt content with name** option to encrypt the exported file.

Other systems can import and run the encrypted file, but the importer cannot edit the file. The encrypted file content is read-only.

- 5 Select a location in which to save the action file and click **Save**.

The action is exported to a local file.

What to do next

You can import the action and use it in workflows and scripts.

Import an Action

You can import actions and use them as building blocks in workflows, Web views, and scripts.

Procedure

- 1 Click the **Actions** view in the Orchestrator client.
- 2 Expand the root of the actions hierarchical list and navigate to the module in which you want to import the action.
- 3 Right-click the module and select **Import action**.
- 4 Select a file with the `.action` extension and click **Open**.

The imported action appears in the actions library.

What to do next

You can use the action in workflows and scripts.

Move an Action

To reorder actions in the actions hierarchical list, or organize your scripts in a different way, move an action to another module.

Procedure

- 1 Click the **Actions** view in the Orchestrator client.
- 2 Expand the root of the actions hierarchical list and navigate to the action to relocate.
- 3 Right-click the action and select **Move this action**.
- 4 Select a location in which to save the action file and click **Save**.

The action is moved to the new module.



CAUTION Action referencing is based on the action module name and action name. Make sure that all elements that reference this action are still valid after you move the action.

What to do next

Find all workflows and packages that implement the relocated action.

Find Elements That Implement an Action

If you edit an action and change its behavior, you might inadvertently break a workflow or application that implements that action. Orchestrator provides a function to find all of the actions, workflows, or packages that implement a given element. You can check whether modifying the element affects the operation of other elements.

IMPORTANT The **Find Elements that Use this Element** function checks all packages, workflows, and policies, but it does not check in scripts. Consequently, modifying an action might affect an element that calls this action in a script that the **Find Elements that Use this Element** function did not identify.

Procedure

- 1 Click the **Actions** view in the Orchestrator client.
- 2 Expand the nodes of the actions hierarchical list to navigate to a given action.
- 3 Right-click the action and select **Find Elements that Use this Element**.
A dialog box shows all of the elements, such as workflows or packages, that implement this action.
- 4 Double-click an element in the list of results to show that element in the Orchestrator client.

You located all of the elements that implement a given action.

What to do next

You can check whether modifying this element affects any other elements.

Managing Tasks

You use a task to schedule a workflow once, or multiple times using a recurrence pattern.

This chapter includes the following topics:

- [“The Tasks View,”](#) on page 25
- [“Create a Task,”](#) on page 25
- [“Edit the Task Recurrence Pattern,”](#) on page 26

The Tasks View

The **Tasks** view in the Orchestrator client displays a list of all scheduled workflows in the system. The workflows are sorted by name or date, together with their status. You can use the **Tasks** view to create, edit, suspend, resume, and cancel scheduled workflows.

When you select a task in the list, the **Tasks** view presents the following tabs that display details about that particular task.

General	Displays general information about the task, including name, start behavior, description, start date, startup user, the name of the scheduled workflow, and a list of the input values for the workflow.
Recurrence	Displays details about the recurrence pattern of the task.
Executions	Displays details about the different runs of a particular scheduled workflow. This information includes the workflow's running status, start and end date, and the user who started it. When you cancel a task, its log information is removed from the system. When you suspend a task, the log information is kept.
Permissions	Displays the permissions accorded to users or groups of users to interact with the workflow. The possible permissions are View, Execute, Inspect, Edit, and Admin.

Create a Task

You can schedule a workflow from the Orchestrator client **Tasks** or **Workflows** views. The user credential that starts the workflow is the same as the credential that schedules the task.

Prerequisites

You must be granted the **Execute** privilege to create a task.

Procedure

- 1 Click the **Tasks** view in the Orchestrator client.
- 2 From the drop-down menu, select **Create task**.
- 3 (Optional) Select **Create task as** to schedule a task using another user's credentials.
- 4 Search for the workflow to schedule.
- 5 Right-click the workflow and click **Select**.
The input parameters dialog box opens.
- 6 Click the **Execution date and time** value's **Not set** link.
A calendar appears.
- 7 Select the start date and time for the workflow and click **OK**.
- 8 From the **Recurrence** drop-down menu, select the recurrence pattern.
- 9 (Optional) Click the **Recurrence end date** value's **Not Set** link and set an end time and date for the workflow.
- 10 Enter the necessary information into the input parameters dialog box.
- 11 Click **Submit** to schedule the workflow.

The scheduled workflow is listed on the **Tasks** view.

What to do next

You can monitor the scheduled workflow and delete the task when it is complete.

Edit the Task Recurrence Pattern

A recurrence pattern is used to specify the way in which a given workflow is scheduled. You can edit the recurrence pattern of a task from the **Tasks** view.

Prerequisites

You must have created a recurrent task.

Procedure

- 1 Click the **Tasks** view in the Orchestrator client.
- 2 Right-click the scheduled task whose recurrence pattern you want to edit and select **Edit**.
- 3 Click the **Recurrence** tab.
- 4 From the drop-down menu, select the recurrence pattern.
The display changes according to the selected pattern.
- 5 You can add an unlimited number of entries to the pattern.
Each entry can be edited.
- 6 Click **Save and close** when you are finished.

The new recurrence pattern for the scheduled workflow appears on the **Recurrence** tab.

What to do next

You can view details about the different runs of the scheduled workflow on the **Executions** tab.

Policies

Policies are event triggers that monitor the activity of the system. Policies run predefined events in response to changes in the status or performance of certain defined objects.

Policies are a series of rules, gauges, thresholds and event filters that run certain workflows when specific predefined events occur. Orchestrator constantly evaluates the policy rules as long as the policy is running. You can implement policy gauges and thresholds that monitor the behavior of vCenter Server objects of the VC:HostSystem and VC:VirtualMachine types.

Orchestrator defines the following types of policy:

- | | |
|-------------------------|---|
| Policy Templates | Master policies. Policy templates are not linked to real objects. They are abstract sets of rules that define the behavior to implement if a certain abstract event occurs. You can see existing policy templates and create templates in the Policy Templates view in the Orchestrator client. |
| Policies | Policies are instances of a template or standalone event triggers that are linked to real objects, and that are triggered by real-life events. You can also create policies without using a template. You can see existing policies and create policies in the Policies view in the Orchestrator client. |

You can organize policy templates into categories, for easier navigation.

Using Packages

Packages are the vehicle for transporting content from one Orchestrator server to another. Packages can contain workflows, actions, policies, Web views, configurations, or resources.

When you add an element to a package, 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, 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 administrator can decide whether to import the package, or choose specific elements to import.

Packages feature digital rights management to control how the receiving server can use the content of the package. Orchestrator signs packages and encrypts the packages for data protection. Packages use X509 certificates to monitor which users export and redistribute elements.

This chapter includes the following topics:

- [“The Packages View,”](#) on page 29
- [“Create a Package,”](#) on page 30
- [“Set User Permissions on a Package,”](#) on page 31
- [“Export a Package,”](#) on page 31
- [“Import a Package,”](#) on page 32
- [“Get and Synchronize a Remote Package,”](#) on page 33

The Packages View

The **Packages** view in the Orchestrator client interface allows you to add, import, export, and synchronize packages.

The **Packages** view consists of a set of tabs that show different types of information about a particular package. In edit package mode, you can insert and remove elements on each tab.

General	Displays general information about the package, including its name, its legal owner, and a description.
Workflows	Displays all the workflows the selected package contains.
Policies	Displays the policy templates associated with the workflows in the selected package.
Actions	Displays the actions used in the workflows in the selected package.
Web View	Displays the Web views that the selected package contains.

Configurations	Displays the configuration elements included in the package.
Resources	Displays the external resources embedded in the selected package.
Used plug-ins	Displays information about the plug-ins associated with the selected package. Plug-ins can have one or more packages associated with them.
Permissions	Displays the permissions accorded to users or groups of users to interact with the package. The possible permissions are View, Execute, Inspect, Edit, and Admin.

Create a Package

You export workflows, policies, actions, plug-ins, resources, Web views, and configuration elements in packages.

Prerequisites

You must have elements such as workflows, actions, and policies to add to a package.

Procedure

- 1 Click the **Packages** view in the Orchestrator client.
- 2 Click the menu button in the title bar of the **Packages** hierarchical list and select **Add package**.
- 3 Provide a name for the package in the dialog box that opens and click **OK**.

The naming convention for packages is `<domain.your_company>.category.<package_name>`. For example, `com.vmware.mycategory.mypackage`.

- 4 Right-click the package and select **Edit**.
The package editor opens.
- 5 Add a description for the package in the **General** tab.
- 6 Click the **Workflows** tab to add workflows to the package.
 - Click **Insert Workflows (list search)** to search for and select workflows in a selection dialog box.
 - Click **Insert Workflows (tree browsing)** to browse and select workflows in a hierarchical list.
- 7 Click the **Policies, Actions, Configurations, and Resources** tabs to add policy templates, actions, configuration elements, and resource elements to the package.
- 8 Click **Insert Webview** in the **Web View** tab to add Web views to the package.
- 9 Click **Insert used plug-in** in the **Used plug-ins** tab to add plug-ins to the package.

You added the required elements to the package.

What to do next

You must set the user permissions for this package.

Set User Permissions on a Package

You set different levels of permission on a package to limit the access that different users or user groups can have to that package.

You select the different users and user groups for which to set permissions from the users and user groups in the Orchestrator LDAP server. Orchestrator defines levels of permissions that you can apply to users or groups.

View	The user can view the elements in the package, but cannot view the schemas or scripting.
Inspect	The user can view the elements in the package, including the schemas and scripting.
Execute	The user can run the elements in the package.
Edit	The user can edit the elements in the package.
Admin	The user can set permissions on the elements in the package.

Prerequisites

You must have created a package, opened it for editing in the package editor, and added to it the necessary elements.

Procedure

- 1 Click the **Permissions** tab in the package editor.
- 2 Click the **Add access rights** link to define permissions for a new user or user group.
- 3 Search for a user or user group by entering text in the **Search** text box.

The search results show all the users and user groups from the Orchestrator LDAP server that match the search.

- 4 Select a user or user group and click **OK**.
- 5 Right-click the user and select **Add access rights**.
- 6 Check the appropriate check boxes to set the level of permissions for this user and click **OK**.

The levels of permissions are not cumulative. To grant a user permission to view the elements, inspect the schema and scripting, run and edit the elements, and change the permissions, you must check all check boxes.

- 7 Click **Save and Close** to exit the package editor.

You created a package and set the appropriate user permissions.

Export a Package

You can export a package and reuse its content on another Orchestrator server. The system adds the certificates for all of the elements that the exported package contains. When the package is imported into another server, these certificates are also imported.

Prerequisites

You must have created a package and added to it the necessary elements.

Procedure

- 1 Click the **Packages** view in the Orchestrator client.
- 2 Right-click the package to export and select **Export package**.
- 3 Browse to select a location in which to save the package and click **Open**.
- 4 (Optional) Click **Add target certificate** to sign the package.
 - a In the list of certificates, select the certificate to use for the exported package.
 - b Click **Select**.
- 5 (Optional) To impose restrictions on the exported package, deselect any of the following options.

Option	Description
View content	When selected, the importer of the package is allowed to view the JavaScript of the elements contained in the package.
Re-Packageable	When selected, the importer of the package is allowed to redistribute the elements contained in the package.
Edit element	When selected, the importer of the package is allowed to modify the elements contained in the package.

- 6 (Optional) Deselect the **Export version history** check box if you do not want to export the version history of the package.
- 7 Click **Save**.

The package is exported.

What to do next

You can use all of the workflows, actions, policies, and Web views from the exported package on the new Orchestrator server.

Import a Package

To reuse workflows, actions, policies, and Web views from one Orchestrator server on another server, you can import them as a package.

IMPORTANT Packages that Orchestrator 3.2 generates are upwardly compatible with Orchestrator 4.0. You can import a package from an Orchestrator 3.2 server into an Orchestrator 4.0 server. Packages from Orchestrator 4.0 are not backwards compatible with Orchestrator 3.2. You cannot import into an Orchestrator 3.2 server a package that an Orchestrator 4.0 server generates.

Prerequisites

On the remote server, you must have created a package and added to it the necessary elements.

Procedure

- 1 Click the **Packages** view in the Orchestrator client.
- 2 From the drop-down menu, select **Import package**.
- 3 Browse to select the package to import and click **Open**.
Certificate information about the exporter appears.
- 4 Review the package import details and select **Import** or **Import and trust provider**.

The Import package view appears. If the imported package element's version is later than the server version, the system selects the element for import.

- 5 (Optional) Deselect the elements you do not want to import.
- 6 Click **Import checked elements**.

The imported package appears in the list of packages.

What to do next

You can use all of the workflows, actions, policies, and Web views from the imported package as new building blocks on your Orchestrator server.

Get and Synchronize a Remote Package

The **Packages** view provides a way to synchronize a package on one Orchestrator server with a package on another server.

If a package already exists on the local server, use the **Synchronize** option. If you want to retrieve a package from a remote server, use the **Get remote package** option.

Synchronizing packages is the only way to be sure to obtain all the elements from the remote server. If you synchronize individual elements, Orchestrator only synchronizes elements that already exist in the local server. To obtain any new elements from the remote server, you must synchronize the package that contains those elements.

Procedure

- 1 Click the **Packages** view in the Orchestrator client.
- 2 Right-click the package to synchronize and select **Synchronize**.
- 3 Log in to the server.

The Orchestrator Synchronization dialog box opens. It displays the differences between the package elements.

- 4 View the comparison between the local and remote package elements, click **Synchronize** and select an option.

Option	Description
none	Local and remote elements have the same version number. No synchronization is required.
commit	The version of the local element is later. The remote element is overwritten.
update	The version of the remote element is later. The local element is updated. If a local element does not exist, you can delete the remote element.
merge	The local and remote packages are overwritten with a merged list of references. The referenced elements remain unchanged.

NOTE If the remote server does not recognize your certificate, you cannot commit elements.

The synchronized package is reloaded.

What to do next

You can use the updated package content in workflows, actions, policies, and Web views.

Default Orchestrator Plug-Ins

Orchestrator includes a collection of plug-ins. Each plug-in exposes an external product API to the Orchestrator platform. Plug-ins provide inventory classes, extend the scripting engine with new object types, and publish notification events from the external system. Each plug-in also provides a library of workflows that represents the typical use cases of the integrated product in an automated fashion.

For a list of plug-ins installed by default with Orchestrator and their basic functions, see [Table 8-1](#).

For a list of plug-ins available as add-ons and separate downloads, see [Table 8-2](#).

Table 8-1. Plug-Ins Installed by Default

Plug-In	Purpose	Configuration	Input Types	Scripting Objects	Inventory
vCenter 4.0	Provides access to the vCenter 4.0 Server API, so that you can incorporate all of the vCenter Server functions into the management processes that you use Orchestrator to automate.	See <i>vCenter Orchestrator Installation and Configuration Guide, Configure the vCenter 4.0 Plug-In</i> .	See <i>vCenter Orchestrator Developer's Guide, Using the Orchestrator API</i> .	See <i>vCenter Orchestrator Developer's Guide, Using the Orchestrator API</i> .	Returns all objects.
vCO Library	Provides workflows that act as templates for customization and automation of client processes. The workflow library includes templates for lifecycle management, provisioning, disaster recovery, hot backup, and many other standard processes. Sources of library processes are accessible and can be copied and edited.	None	See <i>vCenter Orchestrator Developer's Guide, Using the Orchestrator API</i> .	See <i>vCenter Orchestrator Developer's Guide, Using the Orchestrator API</i> .	Stores nothing in the inventory.
Mail	Uses Simple Mail Transfer Protocol (SMTP) to send email.	Default values for the EmailMessage object to use. See <i>vCenter Orchestrator Installation and Configuration Guide, Configure the SSH Plug-In</i> .	None	EmailMessage	Stores nothing in the inventory.

Table 8-1. Plug-Ins Installed by Default (Continued)

Plug-In	Purpose	Configuration	Input Types	Scripting Objects	Inventory
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. Optionally, the SSH plug-in can provide remote file system browsing directly in the vCO Inventory.	See <i>vCenter Orchestrator Installation and Configuration Guide, Define the Default SMTP Connection</i> .	File Folder RootFolder SshConnection	KeyPairManager SSHCommand SSHFile SSHFolder SSHSession	Stores nothing in the inventory.
WebOperator	A demonstration Web interface that allows you to access the workflows in the Orchestrator library and interact with them across a network.	None	None	None	Stores nothing in the inventory.
Enumeration	Allows the creation of common enumerated types.	None	Enumeration	None	Stores nothing in the inventory.
Net	Wrapper to Jakarta Apache Commons Net Library. Provides implementation of Telnet, FTP, and POP3. The POP3 part allows reading email. In combination with the Mail plug-in, the Net plug-in provides full email sending and receiving capabilities.	None	None	FTPClient POP3Client POP3Message TelnetClient	Stores nothing in the inventory.
XML	A complete Document Object Model (DOM) XML parser that you can implement in automated processes.	None	None	XMLDocument XMLElement XMLManager XMLNameNodeMap XMLNode XMLNodeList	Stores nothing in the inventory.

Table 8-1. Plug-Ins Installed by Default (Continued)

Plug-In	Purpose	Configuration	Input Types	Scripting Objects	Inventory
Database	Java Database Connectivity (JDBC) API 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.	None	None	Connection JDBCConnection PreparedStatement ResultSet	Stores nothing in the inventory.
Refactoring	Provides a set of workflows to help you refactor workflows and applications built for VMware Infrastructure 3.5, so that they run on vCenter Server 4.0.	None	None	RefactorDescription RefactorProcess	Stores nothing in the inventory.

Table 8-2. Other Plug-Ins

Plug-In	Remark	Purpose	Configuration	Inventory
Microsoft	Available as a separate download	Provides access to Windows Management Instrumentation (WMI) and reading and writing to Microsoft Active Directory.	In Orchestrator configuration interface, import settings from the LDAP tab or use different LDAP settings for WMI. <i>See vCenter Orchestrator Installation and Configuration Guide, Configuring LDAP Settings.</i>	Computers, ForeignSecurityPrincipals, Program Data, System, Users, Domain Controllers
VMware Infrastructure 3	Available as a separate download	Backward compatibility.	<i>See vCenter Orchestrator Installation and Configuration Guide, Configure the vCenter 4.0 Plug-In.</i>	Returns all objects.
VMware Infrastructure 3.5	Add-on located in <Install_Directory>/extras/plugins.	Backward compatibility.	<i>See vCenter Orchestrator Installation and Configuration Guide, Configure the vCenter 4.0 Plug-In.</i>	Returns all objects.

Modify the Number of Objects a Plug-In Search Obtains

9

By default, using the Orchestrator client to search for objects through a plug-in returns 20 objects at a time. You can modify the plug-in configuration file to increase the number of objects that are returned.

Prerequisites

You must have installed a plug-in in the Orchestrator server.

Procedure

- 1 Navigate to the plug-in configuration folder on the Orchestrator server system.

On Windows: <Install_Directory>\VMware\Orchestrator\app-server\server\vmo\conf\plugins

This folder contains an XML configuration file for each plug-in you have installed in the Orchestrator server.

- 2 Open the XML configuration file of the plug-in for which you want to change the number of search results.
- 3 Add the following line to the XML configuration file for the plug-in.

```
<entry key="ch.dunes.database.fetch-limit">50</entry>
```

This line sets the number of search results to return to 50.

- 4 Save the XML configuration file.
- 5 (Optional) Repeat [Step 2](#) through [Step 4](#) for each plug-in to modify.
- 6 Restart the Orchestrator server.

You increased the number of search results Orchestrator displays for particular plug-ins.

Access Java Classes in JavaScript

By default, Orchestrator restricts JavaScript access to a limited set of Java classes. If you require JavaScript access to a wider range of Java classes, you must set an Orchestrator system property to grant this access.

Allowing the JavaScript engine full access to the Java virtual machine (JVM) presents potential security issues. Malformed or malicious scripts might have access to all of the system components to which the user who runs the Orchestrator server has access. Consequently, by default the Orchestrator JavaScript engine can access only the classes in the `java.util.*` package.

If you require JavaScript access to classes outside of the `java.util.*` package, you can list in a configuration file the Java packages to which to grant JavaScript access. You then set the `com.vmware.scripting.rhino-class-shutter-file` system property to point to this file.

Procedure

- 1 Create a text configuration file to store the list of Java packages to which to grant JavaScript access.

For example, to grant JavaScript access to all the classes in the `java.net` package and to the `java.lang.Object` class, you add the following content to the file.

```
java.net.*
java.lang.Object
```

- 2 Save the configuration file with an appropriate name and in an appropriate place.
- 3 Open the `vmo.properties` system properties file.

The `vmo.properties` file is in the following location:

- `<install-directory>\VMware\Orchestrator\app-server\server\vmo\conf` if you installed the standalone version of Orchestrator.
- `<install-directory>\VMware\Infrastructure\Orchestrator\app-server\server\vmo\conf` if the vCenter Server installer installed Orchestrator.

- 4 Set the `com.vmware.scripting.rhino-class-shutter-file` system property by adding the following line to the `vmo.properties` file.

```
com.vmware.scripting.rhino-class-shutter-file=path_to_your_configuration_file
```

- 5 Save the `vmo.properties` file.
- 6 Restart the Orchestrator server.

The JavaScript engine has access to the Java classes that you specified.

Maintenance and Recovery

The **Troubleshooting** tab in the Orchestrator configuration interface allows you to perform several bulk operations related to workflows and tasks. You can use the **Troubleshooting** tab to globally reset the server and remove all traces of previous runs.

[Table 11-1](#) lists the possible bulk operations.

IMPORTANT Before you click a troubleshooting option, make sure the vCO server is stopped.

Table 11-1. Troubleshooting Options

Action	Description
Cancel all running workflows	Marks all running workflows as cancelled in the database, which prevents the server from restarting the workflows on the next reboot. Allows Orchestrator to exit infinite loops.
Delete all workflow executions	Deletes all tokens with a database script.
Suspend all scheduled tasks	Cancels all scheduled tasks, but does not stop or remove its associated workflow runs.
Clean all server temporary files	Cleans all temporary files that the JBoss server uses that ensure the server persistency. The JBoss server is the application server that underlies the Orchestrator server.
Force plug-in reinstallation when server starts	Used when a plug-in is changed to make sure it is correctly updated on the next server start. NOTE If you change the Orchestrator database after you configure and install the default plug-ins, you must force plug-in reinstallation. Forcing plug-in reinstallation deletes the <code><Install_Directory>\app-server\server\vm\plugins_VSOPuginInstallationVersion.xml</code> file that holds the version of the plug-ins already installed and forces plug-in reinstallation. The plug-in is reinstalled with its original content. Any changes will be lost.

Change the Size of Server Logs

If a server log regenerates multiple times a day, it becomes difficult to determine what causes problems. To prevent this, you can change the default size of the server log. The default size of the server log is 5MB.

Procedure

- 1 Locate the `<Install_Directory>\app-server\server\vm\conf\log4j.xml` file.
- 2 In the `log4j.xml` file, locate the following code block:

```
<appender class="org.jboss.logging.appender.RollingFileAppender" name="FILE">
  <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>
  <param name="File" value="${jboss.server.home.dir}/log/server.log"/>
  <param name="Append" value="true"/>
```

```
<!-- Rollover at 5MB and allow 4 rollover files -->
<param name="MaxFileSize" value="5120KB"/>
<param name="MaxBackupIndex" value="4"/>

<layout class="org.apache.log4j.PatternLayout">
  <!-- The default pattern: Date Priority [Category] Message\n -->
  <param name="ConversionPattern" value="%d{yyyy-MM-dd HH:mm:ss.SSSZ} %-5p [%c{1}] %m%n"/>
</layout>
</appender>
```

3 Change the following lines:

```
<param name="MaxFileSize" value="5120KB"/>
<param name="MaxBackupIndex" value="4"/>
```

The `MaxFileSize` parameter controls the size of the log file, and the `MaxBackupIndex` parameter controls the number of files for the rollover.

NOTE Before you save the file, make sure it does not contain typos. If the file contains typos, the logs will be lost.

The system reads this file dynamically. You do not need to reboot the server.

Index

A

actions
 adding 22
 creating 22
 duplicating 22
 encrypting 23
 exporting 23
 finding elements that implement 24
 importing 23
 moving 23
 referencing 23
 relocating 23
Actions view 21
attributes 15

C

check-pointing 7
configuration elements 15

I

icons, workflow 13
input parameters 15

J

JavaScript 41

M

My Orchestrator view 11

O

Orchestrator API 21
Orchestrator architecture 9
Orchestrator client, login 11
output parameters 15

P

packages
 create 30
 digital rights management 29
 exporting 31
 getting 33
 importing 32
 permissions 31
 restricting reuse 31
 signature 29

 signing 31
 synchronizing 33

Packages view 29
parameters 15
persistence 7
plug-ins, searching 39
policies 27
policy engine 7
policy templates 27

R

recurrent tasks 26

S

schema 15, 16
scripting
 access to Java classes 41
 shutter system property 41
scripting engine 7
security 7
server log 43
standard workflows 14
system properties 41

T

tasks
 adding 25
 creating 25
Tasks view 25
token 15
Troubleshooting options
 cancel executions 43
 clean directories 43
 remove executions 43
 reset current version 43
 suspend tasks 43

U

user roles 8

V

versioning 7

W

workflow attributes 15

- workflow engine **7**
- workflow parameters **15**
- workflow token **16**
- workflow token attributes **16**
- workflow token states **17**
- workflows
 - icon **13**
 - library **14**
- permissions **18**
- recurrence **26**
- running **19**
- scheduling **25**
- standard **14**
- Workflows view **13, 14**