

Using the vCenter Orchestrator Plug-In for vCenter Update Manager

vCenter Orchestrator 4.1
vCenter Update Manager 4.1

This document supports the version of each product listed and supports all subsequent versions until the document is replaced by a new edition. To check for more recent editions of this document, see <http://www.vmware.com/support/pubs>.

EN-000614-00

vmware[®]

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

Copyright © 2011 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 is a registered trademark or trademark 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

Using the vCenter Orchestrator Plug-In for vCenter Update Manager	5
1 Introduction to the VMware vCenter Orchestrator Plug-In for vCenter Update Manager	7
Update Manager Plug-In Components	7
Role of vCenter Orchestrator with the Update Manager Plug-In	7
Plug-In Interaction with Update Manager	7
Installing and Configuring the Update Manager Plug-In	8
Functional Prerequisites for the Update Manager Plug-In	8
Install the Update Manager Plug-In	8
Configure the Update Manager Plug-In	9
2 Update Manager Plug-In Scripting API	11
Access the Update Manager Plug-In API	11
Update Manager Plug-In API Types	12
VUM:Baseline Type	12
VUM:Baselines Type	12
VUM:Compliance Type	12
VUM:Host Type	12
VUM:Patch Type	13
VUM:PatchBaseline Type	13
VUM:PatchInfo Type	13
VUM:Patches Type	14
VUM:VIInventory Type	14
VUM:VumObjects Type	14
Update Manager Plug-In API Classes	14
VumBaseline Class	14
VumBaselineSearchSpec Class	15
VumCompliance Class	15
VumGuestRemediationOption Class	16
VumHost Class	16
VumHostRemediationOption Class	16
VumObjectManager Class	17
VumPatch Class	19
VumPatchBaseline Class	19
VumPatchDetail Class	20
VumPatchInfo Class	20
VumPatchSearchSpec Class	21
VumVIInventory Class	22
Update Manager Plug-In API Enumerated Types	22

3	Using the Update Manager Plug-In Workflow Library	25
	Using the Update Manager Plug-In Inventory	25
	Access the Update Manager Plug-In Workflow Library	25
	Update Manager Plug-In Standard Workflows	26
	Creating Custom Update Manager Plug-In Workflows	27
	Create a Custom Workflow	27
	Example Workflow Schema and Presentation	28
	Index	31

Using the vCenter Orchestrator Plug-In for vCenter Update Manager

Using the vCenter Orchestrator Plug-In for vCenter Update Manager provides information and instructions about configuring and using the VMware® vCenter Orchestrator plug-in for VMware® vCenter Update Manager.

Intended Audience

This information is intended for anyone who is installing and configuring the plug-in, using the API of the plug-in, and using the workflow library. *Using the vCenter Orchestrator Plug-In for vCenter Update Manager* is written for experienced users who are familiar with virtual machine technology, with Orchestrator workflow development, and with vCenter Update Manager.

For more information about Orchestrator, see http://www.vmware.com/support/pubs/orchestrator_pubs.html.

For more information about Update Manager, see http://www.vmware.com/support/pubs/vum_pubs.html.

Introduction to the VMware vCenter Orchestrator Plug-In for vCenter Update Manager

1

The Update Manager plug-in (VMware vCenter Orchestrator plug-in for vCenter Update Manager) allows interaction between vCenter Orchestrator and vCenter Update Manager. You can use the plug-in to run Orchestrator workflows that automate vCenter Update Manager processes.

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

This chapter includes the following topics:

- [“Update Manager Plug-In Components,”](#) on page 7
- [“Installing and Configuring the Update Manager Plug-In,”](#) on page 8

Update Manager Plug-In Components

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

vCenter Orchestrator and Update Manager provide the platform for the plug-in, and the plug-in provides interaction between those products.

Role of vCenter Orchestrator with the Update Manager Plug-In

You must use the Orchestrator configuration interface to install and configure the Update Manager plug-in. You use the Orchestrator client to run and create workflows and access the plug-in API.

The Update Manager plug-in is powered by vCenter Orchestrator. Orchestrator is a development and process-automation platform that provides a library of extensible workflows to manage the VMware vSphere infrastructure and other technologies.

Orchestrator allows integration with management and administration solutions through its open plug-in architecture. vCenter Update Manager is one example of an administration solution that you can integrate with Orchestrator by using plug-ins.

Plug-In Interaction with Update Manager

You can use the plug-in to run Orchestrator workflows that interact with vCenter Update Manager to perform automated tasks in the vSphere infrastructure.

vCenter Update Manager enables centralized, automated patch and version management for VMware vSphere, and offers support for VMware ESX/ESXi hosts, virtual machines, and virtual appliances.

With the Update Manager plug-in, you can perform the following tasks:

- Upgrade and patch ESX/ESXi hosts.
- Install and update third-party software on hosts.

- Upgrade virtual machine hardware, VMware Tools, and virtual appliances.

Installing and Configuring the Update Manager Plug-In

You must use the Orchestrator configuration interface to install and configure the Update Manager plug-in.

Functional Prerequisites for the Update Manager Plug-In

To be able to install and use the Update Manager plug-in, your system must meet the following product prerequisites.

vCenter Orchestrator

Verify that you have a running instance of Orchestrator. You can log in to the Orchestrator configuration interface at http://orchestrator_server:8282. Version 1.0 of the plug-in works with vCenter Orchestrator 4.1.

For information about setting up Orchestrator, see the *vCenter Orchestrator Installation and Configuration Guide*.

vCenter Update Manager

Verify that you have access to an Update Manager instance. Version 1.0 of the plug-in works with Update Manager version 4.1.

For information about setting up Update Manager, see the *VMware vCenter Update Manager Installation and Administration Guide*.

vCenter Server

In the Orchestrator configuration interface, configure the vCenter Server plug-in to connect to the vCenter Server host with which Update Manager is registered.

Install the Update Manager Plug-In

To be able to use the Update Manager plug-in, you must download the `.vmoapp` file containing the plug-in and install it by using the Orchestrator configuration interface.

Prerequisites

- Verify that you are logged in to the Orchestrator configuration interface at http://orchestrator_server:8282.
- Verify that you have downloaded the `.vmoapp` file from <http://www.vmware.com/products/datacenter-virtualization/vcenter-orchestrator/plugins.html>.

Procedure

- 1 On the **General** tab, click **Install Application**.
- 2 Upload the Update Manager plug-in.
 - a Click the magnifying glass icon.
 - b Select the `.vmoapp` file to install.
 - c Click **Open**.
 - d Click **Install**.

The Update Manager plug-in tab appears in the Orchestrator configuration interface.

- 3 On the **Startup Options** tab, click **Restart service** to complete the plug-in installation.

Configure the Update Manager Plug-In

To be able to use the Update Manager plug-in, you must set the IP address of at least one vCenter Server with which an instance of Update Manager is registered.

Prerequisites

- Verify that Orchestrator is connected to at least one vCenter Server with which Update Manager is registered.
- Verify that you are logged in to the Orchestrator configuration interface at **`http://orchestrator_server:8282`**.
- Verify that Orchestrator is connected to at least one vCenter Server with which Update Manager is registered.

Procedure

- 1 Click **Update Manager**.
- 2 Select a URL address from the **Default vCenter Server host** drop-down menu to specify the default vCenter Server instance to use.

The **Default vCenter Server host** drop-down menu lists the URL addresses of the vCenter Server instances configured to work with separate Update Manager instances.

- 3 (Optional) Select to add an additional host.
- 4 Click **Apply changes**.

What to do next

You can log in to the Orchestrator client application and run workflows on vCenter Server inventory objects.

Update Manager Plug-In Scripting API

The Update Manager plug-in scripting API contains classes, with their respective attributes and methods, that allow interaction between Orchestrator and vCenter Update Manager. You can use the API to develop custom workflows that interact with vCenter Update Manager.

This chapter includes the following topics:

- [“Access the Update Manager Plug-In API,”](#) on page 11
- [“Update Manager Plug-In API Types,”](#) on page 12
- [“Update Manager Plug-In API Classes,”](#) on page 14
- [“Update Manager Plug-In API Enumerated Types,”](#) on page 22

Access the Update Manager Plug-In API

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

Procedure

- 1 Log in to the Orchestrator client as an administrator.
- 2 Access the API Explorer from either the Orchestrator client or from the **Scripting** tabs of the workflow, policy, and action editors.
 - To access the API Explorer from the Orchestrator client, click **Tools > API Explorer** in the Orchestrator client toolbar.
 - To access the API Explorer from the **Scripting** tabs of the workflow, policy, and action editors, click **Search API** on the left.
- 3 To expand the hierarchical list of Update Manager plug-in API objects, double-click the **VUM** module in the left pane.

What to do next

You can copy code from API elements and paste it into scripting boxes. For more information about API scripting, see the *vCenter Orchestrator Developer's Guide*.

Update Manager Plug-In API Types

The Update Manager plug-in exposes JavaScript API types that map to the functionality of the vCenter Update Manager API.

VUM:Baseline Type

The VUM:Baseline type uses the VumBaseline class as its scripting object.

A baseline contains a collection of one or more patches, extensions, service packs, patches, or bug fixes.

The VUM:Baseline type contains the following properties.

- id
- serverUri
- name
- description
- lastUpdateTimeAsStr
- baselineType
- contentType
- targetType

VUM:Baselines Type

The VUM:Baselines type uses the FinderResult class as its scripting object.

The VUM:Baselines type contains no properties.

VUM:Compliance Type

The VUM:Compliance type uses the VumCompliance class as its scripting object.

The VUM:Compliance type contains the following properties.

- baselineId
- status
- entity

VUM:Host Type

The VUM:Host type uses the VumHost class as its scripting object.

The VUM:Host type represents a vCenter Server system with which an Update Manager instance is registered.

The VUM:Host type contains the following properties.

- id
- serverUri
- name
- isActive

VUM:Patch Type

The VUM:Patch type uses the VumPatch class as its scripting object.

The VUM:Patch type is an object that contains information about the patch.

The VUM:Patch type contains the following properties.

- serverUri
- info
- detail

VUM:PatchBaseline Type

The VUM:PatchBaseline type uses the VumPatchBaseline class as its scripting object.

The VUM:PatchBaseline type contains the following properties.

- id
- serverUri
- name
- description
- lastUpdateTimeAsStr
- baselineType
- contentType
- targetType
- inclPatches
- exclPatches
- patchSearchSpec

VUM:PatchInfo Type

The VUM:PatchInfo type uses the VumPatchInfo class as its scripting object.

The VUM:PatchInfo type specifies the most important attributes of a patch.

The VUM:PatchInfo type contains the following properties.

- id
- serverUri
- name
- idByVendor
- vendor
- releaseDate
- severity
- targetType
- impactLevel

- bundleType
- updateType

VUM:Patches Type

The VUM:Patches type uses the FinderResult class as its scripting object.

The VUM:Patches type contains no properties.

VUM:VIInventory Type

The VUM:VIInventory type uses the VumVIInventory class as its scripting object.

The VUM:VIInventory type represents a vSphere object used in the Update Manager plug-in.

The VUM:VIInventory type contains the following properties.

- serverUri
- id
- type
- name

VUM:VumObjects Type

The VUM:VumObjects type uses the FinderResult class as its scripting object.

The VUM:VumObjects type contains no properties.

Update Manager Plug-In API Classes

The Update Manager plug-in exposes JavaScript API classes that map to the functionality of the vCenter Update Manager API.

VumBaseline Class

The VumBaseline class contains attributes that are related to the baselines from the Update Manager plug-in.

The VumBaseline class defines the following JavaScript attributes.

Attribute	Returns	Description
baselineType	BaselineType	The type of the baseline. See the BaselineType enumeration for valid values.
contentType	BaselineContentType	The content type of the baseline. See the BaselineContentType enumeration for valid values.
description	String	The description of the baseline.
id	Integer	The unique identifier of the baseline.
lastUpdateTime	java.util.Calendar	The time of the last update.
lastUpdateTimeAsStr	String	The formatted string for the last update timestamp.
name	String	The name of the baseline.
serverUri	String	The URI of the server.
targetType	TargetType	The target type of the baseline. See the TargetType enumeration for valid values.

VumBaselineSearchSpec Class

The `VumBaselineSearchSpec` class contains attributes and methods related to the baseline search criteria for the Update Manager plug-in.

The `VumBaselineSearchSpec` class defines the `VumBaselineSearchSpec(string):VumBaselineSearchSpec` constructor.

The `VumBaselineSearchSpec` class defines the following JavaScript attributes.

Attribute	Returns	Description
<code>baselineTypes</code>	Array of <code>com.vmware.o11n.vmo.plugin.vmware_update_manager.model.BaselineType</code>	The types of the baselines to retrieve.
<code>contentTypes</code>	Array of <code>com.vmware.o11n.vmo.plugin.vmware_update_manager.model.BaselineContentType</code>	The content types of the baselines to retrieve.
<code>ids</code>	Array of Integer	The unique identifiers of the baselines to retrieve.
<code>nameExpressions</code>	Array of String	The names or wildcard name expressions of the baselines to retrieve.
<code>patch</code>	<code>com.vmware.o11n.vmo.plugin.vmware_update_manager.model.patch.PatchInfo</code>	Retrieves baselines containing a particular patch.
<code>targetTypes</code>	Array of <code>com.vmware.o11n.vmo.plugin.vmware_update_manager.model.TargetType</code>	The target types of the baselines to retrieve.
<code>viInventories</code>	Array of <code>com.vmware.o11n.vmo.plugin.vmware_update_manager.model.VIInventory</code>	Specifies vSphere objects to which baselines should be attached.

The `VumBaselineSearchSpec` class defines the following methods.

Method	Returns	Description
<code>setBaselineTypesByStringArray(String[])</code>	Void	Utility method to parse and set <code>BaselineType</code> from string value.
<code>setContentTypesByStringArray(String[])</code>	Void	Utility method to parse and set <code>ContentType</code> from string value.
<code>setTargetTypesByStringArray(String[])</code>	Void	Utility method to parse and set <code>TargetType</code> from string value.

VumCompliance Class

The `VumCompliance` class contains attributes related to the compliance of a vSphere object.

The `VumCompliance` class defines the following JavaScript attributes.

Attribute	Returns	Description
<code>baselineId</code>	Integer	The unique identifier of the baseline.
<code>entity</code>	<code>VumVIInventory</code>	The vSphere object whose status is reported.
<code>status</code>	String	The compliance status of the vSphere object.

VumGuestRemediationOption Class

The VumGuestRemediationOption class contains attributes related to configuration for a virtual machine remediation task.

The VumGuestRemediationOption class defines the following JavaScript attributes.

Attribute	Returns	Description
guestCreateSnapshot	Boolean	Takes a snapshot before initializing the remediation process.
guestKeepSnapshotHours	Integer	The time (in hours) to keep the snapshot. If not set, snapshots are never deleted.
guestSnapshotDescription	String	The description of the snapshot. This parameter is valid only if the guestCreateSnapshot parameter is set.
guestSnapshotName	String	The name of the snapshot. This parameter is valid only if the guestCreateSnapshot parameter is set.
guestTakeMemoryDump	Boolean	Indicates whether the guest operating system performs a memory dump while taking the snapshot.

VumHost Class

The VumHost class contains attributes related to the vCenter Server system with which an Update Manager instance is registered.

The VumHost class defines the following JavaScript attributes.

Attribute	Returns	Description
id	Integer	The unique identifier of the vCenter Server host with which Update Manager is registered.
isActive	Boolean	Indicates whether the Update Manager server is accessible.
name	String	The name or URL of the host.
serverUri	String	The URI of the server.

VumHostRemediationOption Class

The VumHostRemediationOption class contains attributes related to the configuration of a host remediation task.

The VumHostRemediationOption class defines the following JavaScript attributes.

Attribute	Returns	Description
clusterDisableDPM	Boolean	Temporarily disables Distributed Power Management (DPM) for the specified clusters. After the remediation process completes, DPM is automatically re-enabled.
clusterDisableFT	Boolean	Temporarily disables Fault Tolerance (FT) for the specified clusters. After the remediation process completes, FT is automatically re-enabled.
clusterDisableHAC	Boolean	Temporarily disables High Availability (HA) for the specified clusters. After the remediation process completes, HA is automatically re-enabled.
clusterEnableParallelRemediation	Boolean	Enables parallel remediation for the specified clusters.
hostDisableMediaDevices	Boolean	Temporarily disables any media devices that might prevent the specified hosts from entering maintenance mode.
hostEnablePXEbootHostPatching	Boolean	Enables PXE booted ESXi hosts patching.

Attribute	Returns	Description
hostFailureAction	String	Specifies an action in case the host cannot be put in maintenance mode. This might happen due to running virtual machines that cannot be automatically migrated to another host. The possible values are <code>FailTask</code> , <code>Retry</code> , <code>PowerOffVMsAndRetry</code> , and <code>SuspendVMsAndRetry</code> .
hostIgnoreThirdPartyDrivers	Boolean	Ignores any third-party software that is incompatible with the upgrade.
hostNumberOfRetries	Integer	Specifies the number of attempts to set the host in maintenance mode.
hostPreRemediationPowerAction	String	Specifies an action before trying to put the host in maintenance mode. The possible values are <code>PowerOffVMs</code> , <code>SuspendVMs</code> , and <code>DoNotChangeVMsPowerState</code> .
hostRetryDelaySeconds	Integer	Specifies the time (in seconds) to wait for the host to enter maintenance mode.

VumObjectManager Class

The `VumObjectManager` class contains methods related to the Update Manager Object Manager Type.

The `VumObjectManager` class defines the following methods.

Method	Returns	Description
attachBaselines	Void	Attaches baselines to the selected vSphere object. The object can be a template, virtual machine, vApp, ESX/ESXi host, folder, cluster, or datacenter. Attaching a baseline to a container object such as a folder or datacenter attaches the baseline to all objects in the container.
createPatchBaseline	VumBaseline	Creates a new patch baseline. Patch baselines can be applied to hosts or virtual machines. Depending on the patch criteria you select, patch baselines can be either dynamic or static (fixed). You can choose the patches that you want to include in the baseline by using the <code>includePatch</code> parameter. You can also use the <code>searchSpec</code> attribute to filter the patches that you want to include in a baseline. You can filter patches by their properties, such as product, vendor, severity, and release date. The patches that have been excluded by using the <code>excludePatch</code> parameter, are never included, even if they correspond to the filter criteria defined by the <code>searchSpec</code> attribute.
createTrigger	Trigger	Creates a trigger used by the asynchronous workflows.
deleteBaseline	Boolean	Deletes the specified baseline. Before the deletion, the baseline is detached from all objects it is attached to.
deleteBaselines	Void	Deletes the specified baselines. Before the deletion, the baselines are detached from all objects they are attached to.
detachBaselines	Void	Detaches baselines from the selected vSphere objects. To detach inherited baselines, you must detach them from the parent object.
downloadPatchesByLanguages	Boolean	Checks for new patches, and if they are available, downloads them to the Update Manager repository.

Method	Returns	Description
downloadPatchesByLanguagesAsync	Void	Asynchronously checks for new patches, and if they are available, downloads them to the Update Manager repository.
exportBaselines	String	Exports baselines in XML format.
exportCompliance	Void	Exports a compliance report to an external file.
formatAsVumObject	String	Formats a finder (dunesUri) from the vSphere plug-in as VIInventory finder.
getBaselines	Array of VumBaseline	Retrieves the baselines specified by the provided parameters. Update Manager provides the following default dynamic patch baselines <ul style="list-style-type: none"> ■ Critical Host Patches (Predefined) ■ Non-Critical Host Patches (Predefined) Update Manager provides the following default upgrade baselines: <ul style="list-style-type: none"> ■ VMware Tools Upgrade to Match Host (Predefined) ■ VM Hardware Upgrade to Match Host (Predefined) ■ VA Upgrade to Latest (Predefined) You cannot edit or delete the default baselines.
getBundleTypes	Array of String	Retrieves all bundle types.
getCompliance	Array of VumCompliance	Retrieves baseline compliance data for the specified vSphere object such as a template, virtual machine, vApp, ESX/ESXi host, folder, cluster, or datacenter. The method returns information about the compliance of the specified object against the baselines that are attached to it. If the object is a container, the method returns compliance data for all objects in the container.
getDefaultVcUri	String	Utility method that returns the default vCenter Server URI for the plug-in as set in its configuration.
getEntities	Array of VumVIInventory	Connects to a vCenter Server and retrieves all objects from it.
getFilteredBaselines	Array of VumBaseline	Retrieves the baselines specified by the provided parameters.
getPatchInstallationImpacts	Array of String	Retrieves the installation impacts of all patches.
getPatchLanguages	Array of String	Retrieves all patch languages.
getPatchVendors	Array of String	Retrieves all patch vendors.
getPatches	Array of VumPatch	Retrieves patches specified by the provided parameters.
getVcServerUri	Array of String	Returns the list of vCenter Server instances that are set in the plug-in configuration.
importBaselines	Array of VumBaselines	Imports baselines from the .xml file that is generated by using exportBaselines.
remediate	Boolean	Remediates an inventory object against the specified baselines. You can remediate a template, virtual machine, vApp, ESX/ESXi host, folder, cluster, or datacenter.

Method	Returns	Description
remediateAsync	Array of String	Remediates an inventory object against the specified baselines. Returns an array of vCenter Server task keys. You can remediate a template, virtual machine, vApp, ESX/ESXi host, folder, cluster, or datacenter.
scanInventory	Void	Scans vSphere objects against patches included in the baselines attached to them. The object can be a template, virtual machine, vApp, ESX/ESXi host, folder, cluster, or datacenter. If the objects are of different types, the method starts a separate vCenter Server task for each object.
stage	Boolean	Stages patches. Staging allows you to download patches from the Update Manager server to the ESX/ESXi hosts, without applying the patches immediately. You can stage patches to hosts or container objects such as clusters or datacenters.
stageAsync	Array of String	Stages patches. Returns an array of task keys for all of the started vCenter Server tasks. Staging allows you to download patches from the Update Manager server to the ESX/ESXi hosts, without applying the patches immediately. You can stage patches to hosts or container objects such as clusters or datacenters.
updatePatchBaseline	Void	Modifies the properties of a patch baseline.

VumPatch Class

The `VumPatch` class contains attributes related to the patch details.

The `VumPatch` class defines the following attributes.

Attribute	Returns	Description
detail	VumPatchDetail	Additional details about a patch.
info	VumPatchInfo	Additional information about a patch.
serverUri	String	The URI of the server.

VumPatchBaseline Class

The `VumPatchBaseline` class contains attributes and methods related to the patch baseline for the Update Manager plug-in.

The `VumPatchBaseline` class defines the following JavaScript attributes.

Attribute	Returns	Description
baselineType	BaselineType	Specifies the type of a baseline. See the <code>BaselineType</code> enumeration for possible values.
contentType	BaselineContentType	Specifies the content type of a baseline. See the <code>BaselineContentType</code> enumeration for possible values.
description	String	The description of the baseline.
exclPatches	Array of VumPatchInfo	The patches that are excluded from the baseline.
id	Integer	The unique identifier of the baseline.
inclPatches	Array of VumPatchInfo	The patches that are included in the baseline.
lastUpdateTime	java.util.Calendar	The time of the last update.

Attribute	Returns	Description
lastUpdateTimeAsStr	String	Formatted string for the last update timestamp.
name	String	The name of the baseline.
patchSearchSpec	VumPatchSearchSpec	Search criteria for dynamic patches.
serverUri	String	The URI of the server.
targetType	TargetType	Specifies the target type of the baseline. See the TargetType enumeration for possible values.

The VumPatchBaseline class defines the following method.

Method	Returns	Description
setContenttypeFromString(String):void	Void	Utility method to parse and set the content type from a string value.

VumPatchDetail Class

The VumPatchDetail class contains attributes related to the details about the patches used by the Update Manager plug-in.

The VumPatchDetail class defines the following JavaScript attributes.

Attribute	Returns	Description
affectedComponent	Array of String	The affected sub-components.
binaryPath	String	The path or URL of the installable module if it exists.
bugtraqId	Array of String	The unique identifier in a bug tracking system.
cveId	Array of String	The Common Vulnerabilities and Exposures (CVE) identifiers that this update applies to.
description	String	The description of the patch.
detailsUrl	String	The URL for additional details.
idByVendor	Array of String	The patch ID provided by the vendor.
installFlags	String	The command line switches to install the update.
installSeparately	Boolean	Specifies whether the update requires to be installed separately.
patchSize	Integer	The size of the patch.
status	Integer	The patch status.
vendor	String	The patch vendor.

VumPatchInfo Class

The VumPatchInfo class contains attributes related to the most important attributes of the patches used by the Update Manager plug-in.

The VumPatchInfo class defines the following JavaScript attributes.

Attribute	Returns	Description
bundleType	String	Specifies the patch bundle type. The list of possible values is retrieved with a call to <code>VumObjectManager.getBundleTypes()</code> .
id	Integer	The unique identifier of the patch.
idByVendor	String	The patch ID provided by the vendor.

Attribute	Returns	Description
impactLevel	Array of String	Specifies impact level of the patch on the vSphere object. The list of possible values is retrieved with a call to <code>VumObjectManager.getInstallationImpacts()</code> .
name	String	The name of the patch.
releaseDate	<code>java.util.Calendar</code>	The release date of the patch.
serverUri	String	The URI of the server.
severity	<code>com.vmware.011n.vmo.plugin.vmware_update_manager.model.patch.Severity</code>	Specifies the severity level of the patch. See the <code>Severity</code> enumeration for possible values.
targetType	<code>com.vmware.011n.vmo.plugin.vmware_update_manager.model.TargetType</code>	Specifies the target type of the patch. See the <code>TargetType</code> enumeration for possible values.
updateType	<code>UpdateType</code>	Specifies the update type of the patch. See the <code>UpdateType</code> enumeration for possible values.
vendor	String	Specifies the vendor of the patch.

VumPatchSearchSpec Class

The `VumPatchSearchSpec` class contains attributes and methods related to the patch search criteria for the Update Manager plug-in.

The `VumPatchSearchSpec` class defines the `VumPatchSearchSpec(string):VumPatchSearchSpec` constructor.

The `VumPatchSearchSpec` class defines the following JavaScript attributes.

Attribute	Returns	Description
endDate	<code>java.util.Calendar</code>	The end of the search interval.
failOnManyUpdates	Boolean	Update Manager reports an error if the result set is too big.
language	Array of String	The patch language. The list of possible values is retrieved with a call to <code>VumObjectManager.getPatchLanguages()</code> .
phrase	String	A comma-separated list of search keywords. The match is case insensitive.
product	Array of String	The names of software products to retrieve patches for.
serverUri	String	The URI of the server.
severity	Array of <code>com.vmware.011n.vmo.plugin.vmware_update_manager.model.patch.Severity</code>	The patch severity level. See the <code>Severity</code> enumeration for possible values.
startDate	<code>java.util.Calendar</code>	The beginning of the search interval.
targetType	Array of <code>com.vmware.011n.vmo.plugin.vmware_update_manager.model.TargetType</code>	Specifies the patch target type. See the <code>TargetType</code> enumeration for possible values.
updateType	Array of <code>com.vmware.011n.vmo.plugin.vmware_update_manager.model.patch.UpdateType</code>	The patch update type. See the <code>UpdateType</code> enumeration for possible values.
vendor	Array of String	The vendors of the patches.

The `VumPatchSearchSpec` class defines the following methods.

Method	Returns	Description
setSeverityFromStrings(String[])	Void	Utility method to parse and set severity from a string value.
setTargetTypeFromStrings(String[])	Void	Utility method to parse and set target type from a string value.
setUpdateTypeFromStrings(String[])	Void	Utility method to parse and set update type from a string value.

VumVIInventory Class

The `VumVIInventory` class contains attributes and methods related to the vSphere inventory used in the Update Manager plug-in.

The `VumVIInventory` class defines the `VumVIInventory(string):VumVIInventory` constructor.

The `VumVIInventory` class defines the following JavaScript attributes.

Attribute	Returns	Description
id	String	The unique identifier of the vSphere object.
name	String	The name of the vSphere object.
type	String	The vSphere object type.

Update Manager Plug-In API Enumerated Types

You can use the Update Manager plug-in to implement enumerated types in workflows.

The Update Manager plug-in scripting API contains the following enumerated types. You can use enumerations in workflows.

Enumeration Name	Possible Values	Description
VUM:BaselineContentType	BOTH: Both dynamic and static baselines are included. DYNAMIC: Available patches that meet the specified search criteria are included. The baseline is updated when the set of available patches changes. STATIC: The patches to include are specified manually.	Specifies how patches are selected for inclusion in a baseline.
VUM:BaselineType	Extension: Consists of additional software components, such as device drivers, modules, or enhancements, that are applied to hosts. Patch: Consists of patches. Upgrade: Consists of upgrades.	Specifies the contents of a baseline.
VUM:ComplianceStatus	Compliant: The object is compliant with the patch or baseline. Incompatible: Object compliance cannot be checked against this patch or baseline. NotCompliant: The object is not compliant with the patch or baseline. Unknown: Object compliance status cannot be determined.	Specifies different levels of compliance an object can have.
VUM:ExportType	csv: CSV file format. html: HTML file format. pdf: PDF file format.	Specifies the format of the exported file.

Enumeration Name	Possible Values	Description
VUM:ScanType	HostPatch: Patches for HOST_GENERAL TargetComponent HostThirdParty: Patches for HOST_THIRDPARTY TargetComponent HostUpgrade: Upgrades for HOST_GENERAL TargetComponent VaUpgrade: Upgrades for VA_GENERAL TargetComponent VmHardwareUpgrade: Upgrades for VM_HARDWARE_VERSION TargetComponent VmPatch: Patches for VM_GENERAL TargetComponent VmToolsUpgrade: Upgrades for VM_TOOLS TargetComponent	Specifies the possible ways in which a scan can be performed.
VUM:Severity	Critical Important Low Moderate NotApplicable HostGeneral HostSecurity	Specifies the severity of the issues that a patch addresses.
VUM:TargetType	HOST: ESX/ESXi host VA: Virtual appliance VM: Virtual machine	Specifies the type of target objects a patch can be applied to.
VUM:UpdateType	OperatingSystem: Service packs, base OS updates Other: Everything else, like bug fixes, new products, and so on Security: Vulnerability fixes	Specifies the contents of an update.

Using the Update Manager Plug-In Workflow Library

3

The Update Manager plug-in workflow library contains workflows that you can use to run automated processes related to the management of vSphere objects in the inventory of the vCenter Server with which Update Manager is registered.

You can integrate standard workflows from the workflow library to create custom workflows.

This chapter includes the following topics:

- [“Using the Update Manager Plug-In Inventory,”](#) on page 25
- [“Access the Update Manager Plug-In Workflow Library,”](#) on page 25
- [“Update Manager Plug-In Standard Workflows,”](#) on page 26
- [“Creating Custom Update Manager Plug-In Workflows,”](#) on page 27

Using the Update Manager Plug-In Inventory

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

You can enable the **Use contextual menu in inventory** option to display the workflows that are available for an inventory object. When the option is enabled and you right-click an object in the Orchestrator inventory, all available workflows for the object are displayed.

Access the Update Manager Plug-In Workflow Library

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

Procedure

- 1 Log in to the Orchestrator client as an administrator.
- 2 Click the **Workflows** view in the Orchestrator client.
- 3 In the hierarchical list, select **Library > vCenter Update Manager** and expand the selection.

Update Manager Plug-In Standard Workflows

The vCenter Update Manager workflow category contains a set of standard workflows that cover the most common tasks that you can perform with Update Manager. You can use the workflows as building blocks for creating complex custom workflows. By combining standard workflows, you can automate multistep processes in your vSphere environment.

Workflow	Description
Attach a baseline	Attaches baselines to the selected vSphere object. The object can be a template, virtual machine, vApp, ESX/ESXi host, folder, cluster, or a datacenter. Attaching a baseline to a container object such as a folder or datacenter transitively attaches the baseline to all objects in the container.
Create a patch baseline	Creates a new patch baseline. You can apply patch baselines to hosts or virtual machines. Depending on the patch criteria that you select, patch baselines can be dynamic or static (fixed). You can explicitly select the patches to include in the baseline by using the <code>includePatch</code> parameter. You can also use the <code>searchSpec</code> attribute to filter the patches that you want to include. You can filter by product, vendor, severity, and release date. Patches that have been excluded by using the <code>excludePatch</code> parameter are not included in the baseline, even if they correspond to the filter criteria defined by the <code>searchSpec</code> attribute.
Detach a baseline	Detaches baselines from the selected vSphere inventory objects. To detach inherited baselines, you must detach them from the parent object.
Download all patches	Checks whether new patches and updates are available, and downloads the new patches to the Update Manager repository.
Download all patches asynchronously	Checks whether new patches and updates are available, and downloads the new patches to the Update Manager repository.
Export baselines	Exports baselines in XML format.
Export compliance report	Exports the compliance report to an external file format.
Filter baselines	Filters baselines depending on the provided filter parameters. You can select a baseline manually from the filtered list to include it as a workflow result.
Filter baselines with no user interaction	Filters baselines depending on the provided filter parameters.
Filter patches	Filters patches and allows you to select a subset of the filtered patch for further processing.
Filter patches with no user interaction	Filters patches and allows you to select a subset of the filtered patch for further processing.
Get compliance	Retrieves compliance data for the specified object. The object can be a template, virtual machine, vApp, host, cluster, folder, or datacenter. The workflow returns information about the compliance state of the vSphere object against the baselines that are attached to it. If the vSphere object is a container, you receive compliance data for all objects in the container.
Import baselines	Imports baselines from the <code>.xml</code> file that the Export baselines workflow generates.
Remediate	Remediates an inventory object against the specified baselines. You can remediate vSphere objects such as templates, virtual machines, vApps, hosts, folders, clusters, and datacenters.
Remediate asynchronously	Remediates an inventory object against the specified baselines. You can remediate vSphere objects such as templates, virtual machines, vApps, hosts, folders, clusters, and datacenters. The workflow returns an array of vCenter Server task keys.
Remove baselines	Deletes the specified baselines. Before deletion, the baselines are detached from all vSphere objects that they are attached to.
Scan inventory asynchronously	Scans vSphere objects for applicable patches and updates that are included in the attached baselines. You can scan vSphere objects such as templates, virtual machines, vApps, hosts, folders, clusters, and datacenters. If the objects are of different types, the workflow starts a separate vCenter Server task for each object type.

Workflow	Description
Stage	Stages patches to hosts. Staging patches allows you to download patches from the Update Manager server to the ESX/ESXi hosts, without applying the patches immediately. You can stage patches either to hosts or to container objects such as clusters or datacenters.
Stage asynchronously	Stages patches to hosts. Staging patches allows you to download patches from the Update Manager server to the ESX/ESXi hosts, without applying the patches immediately. You can stage patches to hosts or container objects such as clusters or datacenters. Returns an array with task keys for all of the started vCenter Server tasks.
Update a patch baseline	Modifies the properties of an existing patch baseline.

Creating Custom Update Manager Plug-In Workflows

You can use the Orchestrator client to create custom workflows for the Update Manager plug-in.

In the workflows you create, you can combine standard workflows from the workflow library of the plug-in and add custom elements. For more information about workflow development, see the *vCenter Orchestrator Developer's Guide*.

Create a Custom Workflow

You can create a custom Update Manager plug-in workflow that applies a critical patch to the hosts in your vSphere inventory. You can apply patches to a single host or at the folder, cluster, or datacenter level.

The following scenario shows how to create a custom workflow. The tasks performed in this scenario are standard workflows from the Update Manager plug-in workflow library.

Prerequisites

- Review the information about developing workflows. See the *vCenter Orchestrator Developer's Guide*.
- Review the example workflow schema. See [“Example Workflow Schema and Presentation,”](#) on page 28.
- Verify that you are logged in to the Orchestrator client as an administrator.

Procedure

- 1 Create a new workflow.
For example, you can name the workflow Apply a host patch.
- 2 Use the **Schema** tab in the workflow editor to build the workflow.
- 3 Add the Attach a baseline, Scan inventory asynchronously, and Export compliance report workflows to the schema.
- 4 Link the starting point of the custom workflow to the Attach a baseline workflow.
- 5 Link the Attach a baseline workflow to the Scan inventory asynchronously workflow.
- 6 Link the Scan inventory asynchronously workflow to the Export compliance report workflow.
- 7 Add a User Interaction element.
- 8 Link the Export compliance report workflow to the User Interaction element.
- 9 Add a Decision element.
- 10 Link the User Interaction element to the Decision element.
- 11 Add an End workflow element to the right of the Decision element.

- 12 Link the Decision element to the End workflow element.
If the Decision element returns `false`, the workflow stops.
- 13 Add the Remediate workflow.
- 14 Link the Decision element to the Remediate workflow.
If the Decision element returns `true`, the remediation process begins.
- 15 Add an End workflow element.
- 16 Link the Remediate workflow to the End workflow element.
- 17 Edit the workflow elements.
 - a On the **General** tab, edit the workflow description and attributes.
 - b On the **Inputs** tab, edit the input parameters.
 - c On the **Outputs** tab, edit the output parameters.
 - d On the **Presentation** tab, edit the workflow presentation.
- 18 On the **Schema** tab, validate the workflow.
- 19 Save the workflow.

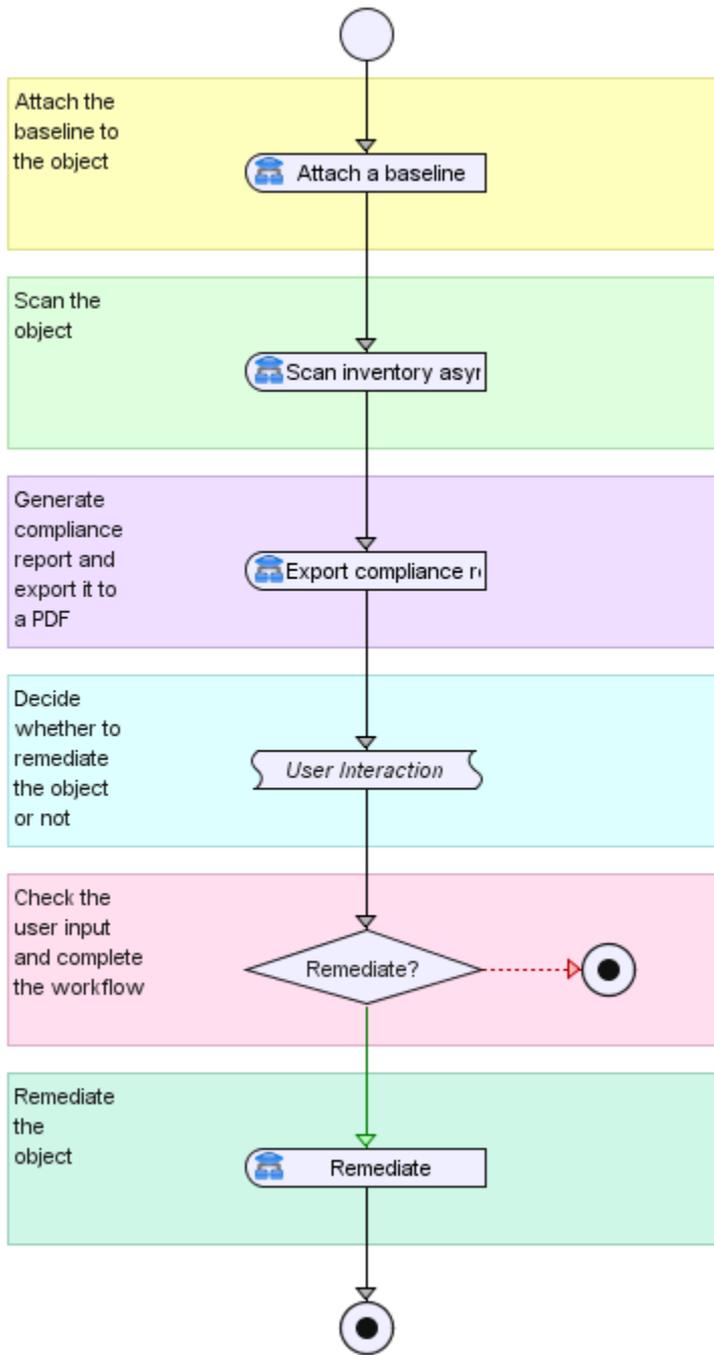
What to do next

You can run the custom workflow, and verify that the patch is applied to the hosts in your vSphere inventory.

Example Workflow Schema and Presentation

You can use the example schema and presentation as a guideline for creating a custom workflow that applies a patch to the hosts in the datacenter.

The following diagram shows an example schema that you can recreate when you build the custom workflow.

Figure 3-1. Example Schema for the Custom Apply a Host Patch Workflow

The compliance report that the Export compliance report workflow generates is stored in the location that you specify. You can access it by using the following JavaScript code.

```
var fr = new FileReader(file);
fr.open();
content = fr.readAll();
fr.close();
```

The following figure shows an example presentation that you can recreate when you build the custom workflow.

Figure 3-2. Example Presentation for the Custom Apply a Host Patch Workflow

Start Workflow : Apply a host patch

Target

vCenter Server

The baseline to attach

vSphere objects

Scan Options

Scan options

Compliance Options

Server to store file

Export file format

Remediation Options

Patches to exclude

Select a vCenter Server to connect to and the baselines to apply to the vSphere objects.

Specify the possible ways in which the scan can be performed.

Filter by compliance status and set the export options. Make sure that you have write permissions for the export directory and that no file with the same name exists.

Specify the patches that are excluded from the baseline.

Index

A

API access **11**

API classes

VumBaseline class **14**

VumBaselineSearchSpec class **15**

VumCompliance class **15**

VumGuestRemediationOption class **16**

VumHost class **16**

VumHostRemediationOption class **16**

VumObjectManager class **17**

VumPatch class **19**

VumPatchBaseline class **19**

VumPatchDetail class **20**

VumPatchInfo class **20**

VumPatchSearchSpec class **21**

VumVIIInventory class **22**

API types

VUM:Baseline type **12**

VUM:Baselines type **12**

VUM:Compliance type **12**

VUM:Host type **12**

VUM:Patch type **13**

VUM:Patches type **14**

VUM:PatchInfo type **13**

VUM:VIIInventory type **14**

VUM:VumObjects type **14**

audience **5**

C

components **7**

configuration **8**

configuration process **9**

custom example presentation **28**

E

Enumerated types **22**

F

functional prerequisites **8**

I

installation **8**

installation process **8**

introduction **7**

Inventory **25**

O

Orchestrator **7**

S

scripting API **11**

U

Update Manager **7**

V

VUM:PatchBaseline type **13**

W

workflow library **25**

workflow library access **25**

workflows

custom **27**

custom example schema **28**

custom scenario example **27**

standard **26**

