

Using the vCenter Orchestrator Plug-In for Microsoft Active Directory

vCenter Orchestrator 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-000585-00

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 Microsoft Active Directory 5

- 1 Introduction to the VMware vCenter Orchestrator Plug-In for Microsoft Active Directory 7**
 - Active Directory Plug-In Components 7
 - Role of vCenter Orchestrator with the Active Directory Plug-In 7
 - Plug-In Interaction with Microsoft Active Directory 7
 - Installing and Configuring the Active Directory Plug-In 8
 - Active Directory Plug-In Functional Prerequisites 8
 - Install the Active Directory Plug-In 8
 - Configure the Active Directory Plug-In 8

- 2 Active Directory Plug-In Scripting API 11**
 - Access the Active Directory Plug-In API 11
 - Active Directory Plug-In API Classes 11
 - AD_Computer Class 11
 - AD_FavoritesManager Class 12
 - AD_Group Class 12
 - AD_OrganizationalUnit Class 13
 - AD_Unknown Class 14
 - AD_User Class 14
 - AD_UserGroup Class 15
 - ActiveDirectory Class 16

- 3 Using the Active Directory Plug-In Workflow Library 17**
 - Using the Active Directory Plug-In Inventory 17
 - Access the Active Directory Plug-In Workflow Library 17
 - Active Directory Plug-In Workflows 18
 - Computer Workflows 18
 - Organizational Unit Workflows 18
 - User Workflows 18
 - User Group Workflows 19
 - Favorites Management Workflows 19

Index 21

Using the vCenter Orchestrator Plug-In for Microsoft Active Directory

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

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 Microsoft Active Directory* is written for experienced users who are familiar with virtual machine technology, with Orchestrator workflow development, and with Microsoft Active Directory.

For more information about Orchestrator, see

http://www.vmware.com/support/pubs/orchestrator_pubs.html.

For more information about Microsoft Active Directory, see

<http://www.microsoft.com/events/series/adaug.aspx> and

[http://technet.microsoft.com/en-us/library/cc782657\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/cc782657(WS.10).aspx).

Introduction to the VMware vCenter Orchestrator Plug-In for Microsoft Active Directory

1

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

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

This chapter includes the following topics:

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

Active Directory Plug-In Components

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

vCenter Orchestrator and Microsoft Active Directory provide the platform for the plug-in, and the plug-in provides interaction between those products.

Role of vCenter Orchestrator with the Active Directory Plug-In

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

The Active Directory 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 vCenter infrastructure and other technologies.

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

Plug-In Interaction with Microsoft Active Directory

You can use the plug-in to run Orchestrator workflows that interact with Microsoft Active Directory and perform automated tasks in the Active Directory infrastructure.

Active Directory is a directory service that stores information about objects on a network and makes this information available to users and network administrators.

Installing and Configuring the Active Directory Plug-In

You must use the Orchestrator configuration interface to install and configure the Active Directory plug-in.

Active Directory Plug-In Functional Prerequisites

To be able to install and use the Active Directory 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 and 4.1.1.

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

Microsoft Active Directory

Verify that you have access to a Microsoft Active Directory instance. Version 1.0 of the plug-in works with Microsoft Active Directory 2003 and 2008.

For information, see the Microsoft Active Directory documentation.

Install the Active Directory Plug-In

To be able to use the Active Directory plug-in, you must download the .vmoapp file containing the plug-in and install it 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 Active Directory plug-in.
 - a Click the magnifying glass icon.
 - b Select the .vmoapp file to install.
 - c Click **Open**.
 - d Click **Install**.The Active Directory 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 Active Directory Plug-In

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

Prerequisites

Verify that you are logged in to the Orchestrator configuration interface at http://orchestrator_server:8282.

Procedure

- 1 Click **Active Directory**.
- 2 (Optional) Leave the **Copy from LDAP configuration** check box selected to use the Orchestrator LDAP configuration settings.
This option is applicable only if you use Active Directory as the Orchestrator LDAP client.
- 3 In the **LDAP host** text box, type the IP address or the DNS name of the host on which Active Directory runs.
- 4 In the **Port** text box, type the value for the look up port of your LDAP server.

NOTE Orchestrator supports Active Directory hierarchical domains structure. If your Domain Controller is configured to use Global Catalog, you must use port 3268. You cannot use the default port 389 to connect to the Global Catalog server.

- 5 In the **Root** text box, type the root element of your LDAP service.
If your domain name is company.org, your root LDAP is **dc=company,dc=org**.
This is the node used to browse your service directory after typing the appropriate credentials. For large service directories, specifying a node in the tree narrows the search and improves performance. For example, rather than searching in the entire directory, you can specify **ou=employees,dc=company,dc=org**. This displays all the users in the Employees group.
- 6 (Optional) Select the **Use SSL** check box to activate encrypted certification for the connection between Orchestrator and Active Directory.
- 7 In the **Default domain** text box, type the default domain.
If your domain name is company.org, type **@company.org**.
- 8 Provide credentials for a user with full browsing rights on Active Directory.
- 9 (Optional) To enable users to log in with a shared user name, select the **Use a shared session** check box and provide credentials for the shared session.
- 10 Click **Apply changes**.

Active Directory Plug-In Scripting API

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

This chapter includes the following topics:

- [“Access the Active Directory Plug-In API,”](#) on page 11
- [“Active Directory Plug-In API Classes,”](#) on page 11

Access the Active Directory Plug-In API

Orchestrator provides an API Explorer to allow you to search the Active Directory 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 Active Directory plug-in API objects, double-click the **AD** 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*.

Active Directory Plug-In API Classes

The Active Directory plug-in exposes JavaScript API classes related to logical unit management.

AD_Computer Class

The `AD_Computer` class contains attributes and methods related to Active Directory computer management.

The `AD_Computer` class defines the following attributes.

Attribute	Returns	Description
allAttribute	Array of Object	Returns all attributes.
distinguishedName	String	Returns the distinguished name of an item.
enabled	Boolean	Allows you to enable or disable a computer.
hostname	String	Remote host name.
id	String	Unique ID of an element.
name	String	Node name.

The AD_Computer class defines the following methods.

Method	Returns	Description
addAttribute(string, Object):void	Void	Adds an attribute.
destroy():void	Void	Destroys an element from the Active Directory permanently.
getArrayAttribute(string):string[]	Array of String	Gets an Active Directory attribute for an array of values.
getAttribute(string):string	String	Gets an Active Directory attribute.
removeAttribute(string):void	Void	Removes an attribute.
rename(string):void	Void	Replaces the name of an attribute with a new one specified as the parameter.
setAttribute(string, Object):void	Void	Changes the value of an existing attribute.

AD_FavoritesManager Class

The AD_FavoritesManager class contains attributes and methods related to the management of favorite Active Directory objects.

The AD_FavoritesManager class defines the following attributes.

Attribute	Returns	Description
id	String	Unique ID of an element.
name	String	Name of an element.

The AD_FavoritesManager class defines the following methods.

Method	Returns	Description
addFavorites(Any):void	Void	Adds an Active Directory element to a list of favorites.
removeFavorites(Any):void	Void	Removes an Active Directory element from the list of favorites.

AD_Group Class

The AD_Group class contains attributes and methods related to Active Directory group management.

The AD_Group class defines the following attributes.

Attribute	Returns	Description
allAttribute	Array of Object	Returns all attributes.
computers	Array of Object	List of all computers.
containers	Array of Object	List of all Group objects.
distinguishedName	String	Returns the distinguished name of an item.

Attribute	Returns	Description
id	String	Unique ID of an element.
organizationUnit	Array of Object	List of all organizational units.
userGroups	Array of Object	List of all user groups.
users	Array of Object	List of all users.

The AD_Group class defines the following methods.

Method	Returns	Description
addAttribute(string, Object):void	Void	Adds an attribute.
createComputer(string, string):void	Void	Creates a new computer and adds it to this container.
createOrganizationalUnit(string):void	Void	Creates a new organizational unit and adds it to this container.
createUser(string, string, string):void	Void	Creates a new user and adds it to this container.
createUserGroup(string):void	Void	Creates a new user group and adds it to this container.
destroy():void	Void	Destroys an element from the Active Directory permanently.
getArrayAttribute(string):string[]	Array of String	Gets an Active Directory attribute for an array of values.
getAttribute(string):string	String	Gets an Active Directory attribute.
removeAttribute(string):void	Void	Removes an attribute.
rename(string):void	Void	Replaces the name of an attribute with a new one specified as the parameter.
setAttribute(string, Object):void	Void	Changes the value of an existing attribute.

AD_OrganizationalUnit Class

The AD_OrganizationalUnit class contains attributes and methods related to Active Directory organizational unit management.

The AD_OrganizationalUnit class defines the following attributes.

Attribute	Returns	Description
allAttribute	Array of Object	Returns all attributes.
computers	Array of Object	List of all computers.
containers	Array of Object	List of all Group objects.
distinguishedName	String	Returns the distinguished name of an item.
id	String	Unique ID of an element.
organizationUnit	Array of Object	List of all organizational units.
userGroups	Array of Object	List of all user groups.
users	Array of Object	List of all users.

The AD_OrganizationalUnit class defines the following methods.

Method	Returns	Description
addAttribute(string, Object):void	Void	Adds an attribute.
createComputer(string, string):void	Void	Creates a new computer and adds it to this container.

Method	Returns	Description
<code>createOrganizationalUnit(string):void</code>	Void	Creates a new organizational unit and adds it to this container.
<code>createUser(string,string,string):void</code>	Void	Creates a new user and adds it to this container.
<code>createUserGroup(string):void</code>	Void	Creates a new user group and adds it to this container.
<code>destroy():void</code>	Void	Destroys an element from the Active Directory permanently.
<code>getArrayAttribute(string):string[]</code>	Array of String	Gets an Active Directory attribute for an array of values.
<code>getAttribute(string):string</code>	String	Gets an Active Directory attribute.
<code>removeAttribute(string):void</code>	Void	Removes an attribute.
<code>rename(string):void</code>	Void	Replaces the name of an attribute with a new one specified as the parameter.
<code>searchComputer(string):AD_Computer</code>	AD_Computer	Searches for a computer by name.
<code>setAttribute(string,Object):void</code>	Void	Changes the value of an existing attribute.

AD_Unknown Class

The AD_Unknown class contains attributes and methods related to the management of unknown Active Directory objects.

The AD_Unknown class defines the following attributes.

Attribute	Returns	Description
<code>allAttribute</code>	Array of Object	Returns all attributes.
<code>distinguishedName</code>	String	Returns the distinguished name of an item.
<code>id</code>	String	Unique ID of an element.

The AD_Unknown class defines the following methods.

Method	Returns	Description
<code>addAttribute(string,Object):void</code>	Void	Adds an attribute.
<code>destroy():void</code>	Void	Destroys an element from the Active Directory permanently.
<code>getArrayAttribute(string):string[]</code>	Array of String	Gets an Active Directory attribute for an array of values.
<code>getAttribute(string):string</code>	String	Gets an Active Directory attribute.
<code>removeAttribute(string):void</code>	Void	Removes an attribute.
<code>rename(string):void</code>	Void	Replaces the name of an attribute with a new one specified as the parameter.
<code>setAttribute(string,Object):void</code>	Void	Changes the value of an existing attribute.

AD_User Class

The AD_User class contains attributes and methods related to Active Directory user management.

The AD_User class defines the following attributes.

Attribute	Returns	Description
accountName	String	Returns the SAM-Account-Name.
allAttribute	Array of Object	Returns all attributes.
distinguishedName	String	Returns the distinguished name of an item.
enabled	Boolean	Allows you to enable or disable a user.
id	String	Unique ID of an element.
memberOf	Array of AD_UserGroup	Returns groups that contain the user.
userPrincipalName	String	User principal name.

The AD_User class defines the following methods.

Method	Returns	Description
addAttribute(string, Object):void	Void	Adds an attribute.
destroy():void	Void	Destroys an element from the Active Directory permanently.
getArrayAttribute(string):string[]	Array of String	Gets an Active Directory attribute for an array of values.
getAttribute(string):string	String	Gets an Active Directory attribute.
removeAttribute(string):void	Void	Removes an attribute.
rename(string):void	Void	Replaces the name of an attribute with a new one specified as the parameter.
setAttribute(string, Object):void	Void	Changes the value of an existing attribute.

AD_UserGroup Class

The AD_UserGroup class contains attributes and methods related to Active Directory user group management.

The AD_UserGroup class defines the following attributes.

Attribute	Returns	Description
allAttribute	Array of Object	Returns all attributes.
computerMember	Array of AD_Computer	Gets the computers that are members of the current group.
distinguishedName	String	Returns the distinguished name of an item.
groupMember	Array of AD_UserGroup	Gets the groups that are members of the current group.
id	String	Unique ID of an element.
memberOf	Array of AD_UserGroup	Gets the groups of which the current group is a member.
userMembers	Array of AD_User	Gets the users that are members of the current group.

The AD_UserGroup class defines the following methods.

Method	Returns	Description
addAttribute(string, Object):void	Void	Adds an attribute.
addElement(Object[]):void	Void	Adds elements to the group.
destroy():void	Void	Destroys an element from the Active Directory permanently.
getArrayAttribute(string):string[]	Array of String	Gets an Active Directory attribute for an array of values.
getAttribute(string):string	String	Gets an Active Directory attribute.
removeAttribute(string):void	Void	Removes an attribute.

Method	Returns	Description
<code>removeElements(Object[]):void</code>	Void	Removes elements from the group.
<code>rename(string):void</code>	Void	Replaces the name of an attribute with a new one specified as the parameter.
<code>setAttribute(string,Object):void</code>	Void	Changes the value of an existing attribute.

ActiveDirectory Class

The `ActiveDirectory` class represents the Active Directory base.

The `ActiveDirectory` class defines the following methods.

Method	Returns	Description
<code>getAllDomainControllerComputers():Object[]</code>	Array of Object	Returns all <code>AD_Computer</code> instances for all subdomain domain controllers.
<code>getComputerAD(string):AD_Computer</code>	<code>AD_Computer</code>	Returns a computer by exact name. Returns <code>null</code> if no match is found.
<code>getComputerADRecursively(string):Object[]</code>	Array of Object	Returns computers recursively for the whole domain tree.
<code>getDC():AD_Computer</code>	<code>AD_Computer</code>	Returns a computer representing the domain controller.
<code>search(string,string):Any[]</code>	Array of Any	Searches the Active Directory for a specific type of object.
<code>searchRecursively(string,string):Object[]</code>	Array of Object	Searches recursively the domain tree of Active Directory for a specific type of object.

Using the Active Directory Plug-In Workflow Library

3

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

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

This chapter includes the following topics:

- [“Using the Active Directory Plug-In Inventory,”](#) on page 17
- [“Access the Active Directory Plug-In Workflow Library,”](#) on page 17
- [“Active Directory Plug-In Workflows,”](#) on page 18

Using the Active Directory Plug-In Inventory

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

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 Active Directory Plug-In Workflow Library

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

Procedure

- 1 Log in to the Orchestrator client as an administrator.
- 2 Click the **Workflows** view.

Option	Action
Access the set of standard workflow categories	In the hierarchical list, select Library > Microsoft > Active Directory and expand the selection.
Access the set of workflows related to the management of favorite objects	In the hierarchical list, select Library > Microsoft > Favorites Management and expand the selection.

Active Directory Plug-In Workflows

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

Computer Workflows

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

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

Workflow Name	Description
Create a computer in a group	Creates an Active Directory computer in a group.
Create a computer in an organizational unit	Creates an Active Directory computer in an organizational unit.
Destroy a computer	Deletes a computer from an Active Directory instance.
Disable a computer	Disables a computer from an Active Directory instance.
Enable a computer	Enables a computer in an Active Directory instance.

Organizational Unit Workflows

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

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

Workflow Name	Description
Create an organizational unit	Creates an organizational unit in an existing organizational unit.
Destroy an organizational unit	Deletes an organizational unit from an Active Directory instance.

User Workflows

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

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

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

Workflow Name	Description
Enable a user	Enables a user in an Active Directory instance.
Remove a user from a user group	Removes a user from a user group.

User Group Workflows

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

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

Workflow Name	Description
Add computers to group members	Adds one or more computers as members of a user group.
Add groups to group members	Adds one or more user groups as members of a user group.
Add users to group members	Adds one or more users as members of a user group.
Create a user group in a group	Creates a new user group in an existing container (group).
Create a user group in an organizational unit	Creates a new user group in an existing container (organizational unit).
Destroy a user group	Deletes a user group from an Active Directory instance.
Remove computers from group members	Removes one or more computers from a user group.
Remove groups from group members	Removes one or more user groups from a user group.
Remove users from group members	Removes one or more users from a user group.

Favorites Management Workflows

The Favorites Management workflow category contains workflows related to the management of favorite Active Directory objects.

You can access these workflows from **Library > Microsoft > Favorites Management**.

Workflow Name	Description
Add a favorite computer	Adds a computer to the list of favorites.
Add a favorite group	Adds a group to the list of favorites.
Add a favorite organizational unit	Adds an organizational unit to the list of favorites.
Remove a favorite computer	Removes a computer from the list of favorites.
Remove a favorite group	Removes a group from the list of favorites.
Remove a favorite organizational unit	Removes an organizational unit from the list of favorites.

You can access the list of favorite objects from the **Inventory** view in the Orchestrator client.

Index

A

Active Directory 7

API access 11

API classes

ActiveDirectory class 16

AD_Computer class 11

AD_FavoritesManager class 12

AD_Group class 12

AD_OrganizationalUnit class 13

AD_Unknown class 14

AD_User class 14

AD_UserGroup class 15

audience 5

C

components 7

configuration 8

configuration process 8

F

functional prerequisites 8

I

installation 8

installation process 8

introduction 7

Inventory 17

O

Orchestrator 7

S

scripting API 11

W

workflow library 17

workflow library access 17

workflows

computer 18

favorites management 19

organizational unit 18

user 18

user group 19

