Active Directory Integration in System Center Operations Manager 2007

A walkthrough configuration of the AD Integration feature and manual agent deployment options

Authors:
Pete Zerger, MCSE (Messaging) | MCTS (OpsMgr 2007) | MCTS (SQL 2005) | MVP-OpsMgr
Anders Bengtsson, MCSE (Security) | MVP-OpsMgr
Raphael Burri , MCTS (OpsMgr 2007)

Version: 2.1
December 2008

Table of Contents

Introduction

Operations Manager 2007 integration with Active Directory is designed to minimize cost and effort in Operations Manager agent administration by allowing an administrator to publish agent configuration details in the Active Directory domain partition related to desired Operations Manager 2007 Management Group membership, as well as desired primary and failover management server settings. The Operations Manager 2007 agent actually issues an LDAP query to its authenticating domain controller at startup to determine which management group and management server(s) it should report to.

It's important to understand that this does not automate the deployment of the agent, but rather automates the configuration of Operations Manager agents that have been deployed by any number of manual installation methods, including installation directly from the product installation media, via group policy, via a systems management platform such as SMS, or even as part of a machine image.

Configuration of Active Directory Integration in Operations Manager 2007 consists of the following high level steps:

1. Run the MOMADAdmin.exe utility with an account that is a member of Domain Admins – This facilitates creation of the containers in the AD that will store the advertised management group settings.
2. Configure agent to management server assignment and failover settings in the Operations Console interface.
3. Deploy agents through the manual installation method of your choice – we'll cover some of these options later in this document.

Running the MOMADAdmin.exe Utility

To create required containers, security groups and ACLs in the directory you will use the MOMADAdmin.exe tool. You will have to run this tool with an account that is member of the Domain Admin security group. It is important to note that MOMADAdmin does NOT need to be run on a domain controller. In the example below we have a domain named noc.momresources.org (NetBIOS name of NOC) and a management group named LAB. 'OpsMgr Administrators' is a global security group containing the user accounts of Operations Manager administrators. NOCRMS01 is our root management server. The following is the MOMADAdmin syntax for configuring Active Directory integration:

Syntax;

MOMADAdmin ManagementGroupName MOMADAdminSecurityGroup {RootManagement | RunAsAccount} Domain

Example:

MomADAdmin.exe lab "noc\OpsMgr Administrators" nocrms01 noc

The command line parameters correspond to the following:

1) Management Group
2) Operations Manager Administrators Security Group
3) Root Management Server | RunAs Account*
4) Name of the AD domain

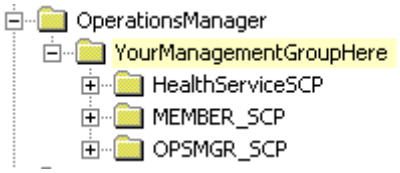
*If the rule is being run under a RunAs profile, this parameter needs to correspond to this account, rather than the RMS.

Where to get it:
You will find MOMADAdmin.exe at your installation source under the **\SupportTools** directory.

What does the MOMADAdmin tool do?

When you run the MOMADAdmin tool, it performs the following actions.

1. Creates a top level container called **OperationsManager** in AD under the root of the specified Domain. Under that container, it creates a management group container - whose name reflects the management group name, as shown in the figure below.



2. Adds the machine account of the root management server to the MOM Admin security group.

3. Adds the MOM Admin security group to the container's ACL with WriteChild access (only with rights to create Security Groups and SCP objects, but not other object classes like user accounts). This allows members of the MOM administrator role and the root management server to manage the objects within the container. Only Domain Administrators have the right to remove or change the OperationsManager and <ManagementGroup> container.

   A **service connection point** is a special object class in Active Directory used to publish information about applications. In this case, the objects will contain information about desired agent settings that can be queried by the agent at startup.

   If the root management server or the Operations Manager administrator's security group is changed you will need to modify objects in this new container. The easiest way would be to delete the container and then run the momadadmin.exe tool again. If you use a RunAs profile instead (as explained later in this document), this will not be necessary.

FYI

For a more detailed explanation of the containers and security groups created by the MOMADAdmin tool, see the following blog post from the MOM Team Blog; http://blogs.technet.com/momteam/archive/2008/01/02/understanding-how-active-directory-integration-feature-works-in-opsmgr-2007.aspx

Configuring Auto Agent Assignment

When you have verified that you have all objects that MOMADAdmin.exe creates in Active Directory, you should then configure auto agent assignment in the Operations Console.

1. From the **Start Menu**, select the SCOM **Operations Console**
2. In the Navigation pane (left)**,** Click **Administration**
3. In the Administration pane, click **Management Servers**
4. In the Action pane, right click a **management server** and select **properties** from the context menu
5. On the **Auto Agent Assignment tab**, click **Add…**
6. On the **Introduction** screen, click **Next**
7. On the **Domain** screen, click select **domain** and an **account** to perform agent assignment with. In most cases you can leave default settings and click **next**
8. On the **Inclusion Criteria** screen, click **Configure…**
9. On the **Find Computer** screen, create your query for machines manage by this management server. In this example we want this management server to manage all computers with a hostname beginning with SR.
10. On the **Find Computer** screen, click the **Advanced** tab
11. On the **Advanced** tab, click **Field** and choose **Computer name**, click **condition** and choose **starts with**, input SR in the value box, click **Add**, click **OK**
12. On the **Inclusion Criteria** screen, the query should appear as follows:
    (&(sAMAccountType=805306369)(objectCategory=computer)(samAccountName=SR*))

    Click **Next**

    If you are into LDAP query you can input it into the box directly without using the menus in the dialog box. You also can use the Member Of attribute and control which agents are managed by which management server with which security groups in Active Directory.

13. On the **Exclusion Criteria** screen, input computer names that fulfill your query but you don't want to include. Click **Next**

14. On the **Agent Failover** screen, configure agent failover. You can either choose to automatically failover to any other management server, or you can manually configure failover to specified management servers. Click **Create**

TIP The automatic failover option only works well when all agents can access all management servers. However, selecting automatic failover may not balance agent load in an optimal way during failure conditions.

The above procedure creates a rule, stored in the Default Management Pack, which is scheduled to run on an hourly basis. The rule creates and populates a number of AD security groups. The membership of these security groups will include all machines, based on auto agent assignment settings, which should connect to the management server. The SCP name is the management server's NetBIOS name with the suffix "_SCP".

The rule also creates 2 security groups with the name of the management server's NetBIOS name, the first one with the suffix "_PrimarySG_<random number>" and the second one "_SecondarySG_<random number>"
You can test your query using Active Directory Users and Computers before using them in Operations Manager. See "testing LDAP filters" in this document.

NOTE: It can take some time (up to one hour) before the necessary objects advertising the desired settings are created, as the agent assignment rule that updates the security groups is scheduled to run every 60 minutes.

Granularity in Agent Assignment

It is also possible to use more complex LDAP queries for a more granular agent assignment process. For example, assignment of agents to management servers could be based on any of the following:

- Computer name
- Computer description
- Computer account security group membership
- Operation system and service pack
- Registered Service Principal Names (SPN)
- Computer account Organizational Unit (OU)

See "LDAP Query Syntax and Samples" included later in this document.

Global Security Settings for Manual Installed Agents

As in MOM 2005, manually installed agents are rejected by default. If you deploy agents by any method other than push-install, agents will be rejected unless the default values in Global Security Settings are updated to either auto-approve manually installed agents, or to accept manually installed agents for review in the Pending Management view.

1. From the **Start Menu**, select the SCOM **Operations Console**
2. In the Navigation pane (left)**,** Click **Administration**
3. In the Administration pane, click **Settings**
4. In the Action pane, right click a **Security** and select **properties** from the context menu
5. On the **General tab**, click **Review new manual agent installations in pending management view**. This setting will let you review all manual agent installations. Click **OK**

Installing Agents to use the Active Directory Integration Feature

Remember that you must logon as a local administrator to install the Operations Manager agent.

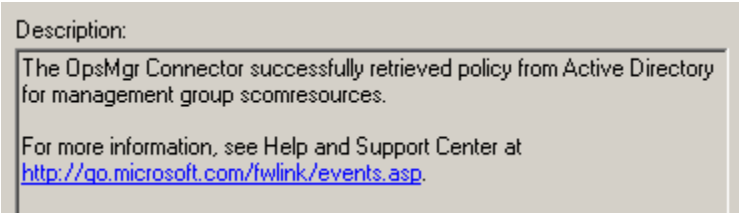
Install an Agent from Command Prompt

1. From the **Start Menu**, start a **command prompt**
2. In the Command Prompt window, change directory to the directory where the MOMagent.msi is stored.
3. In the Command Prompt window, run the following command **MOMAgent.msi /qn USE_SETTINGS_FROM_AD=1 USE_MANUALLY_SPECIFIED_SETTINGS=0**
4. This will install the agent with settings from Active Directory. The /qn turn of all display settings during installation. You can look in the Application Event Log for an event from the MsiInstaller for the result.

You can also run MOMAgent.msi from the command prompt or from Windows Explorer. Then you can input configuration during the wizard. After you choose **Use Management Group Information from**

Active Directory and click next, there will not be any installation settings on the **Ready to Install** screen, this is by design. After the setup is complete the agent will query Active Directory from settings.

When the agent has successfully retrieved configuration from Active Directory, an event like this will be logged on the agent-managed computer.



Applying Hotfixes

If you need to apply hotfixes to agents, they must be installed similarly. MSI transform packages (.msp files) for the agents can be found on any patched management server in the following directory:
C:\Program Files\System Center Operations Manager 2007\AgentManagement

1. Decide which hotfixes have to be applied and copy the .msp files to the agent computers
2. At the command prompt run the following command (you may apply more than one fix in a single command)
   msiexec /p [Full Path to Transform 1].msp;[Full Path to Transform 2].msp /qn

Approve the Manually Installed Agent

After the MSI package has been installed successfully you will have to approve the agent (assuming you selected the “Review manual agent installations” option in Global Settings.

1. From the **Start Menu**, select the SCOM **Operations Console**
2. In the Navigation pane (left)**,** Click **Administration**
3. In the Administration pane, click **Pending Management**
4. In the Action pane, right click a **agent** that are waiting approval, choose **Approve** from the context menu
5. In the Manual Agent Install screen, click **Approve**

Deploy Agent through Group Policy

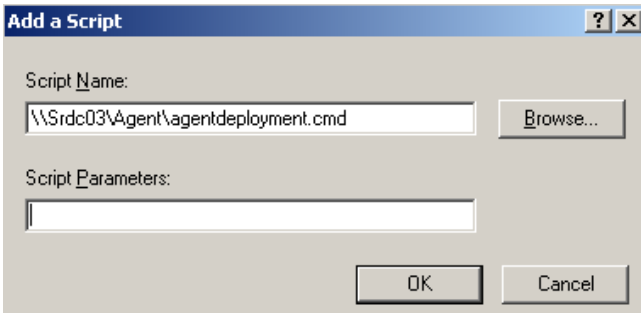
To deploy momagent.msi with a group Policy (GPO), you will need to create a MST file, because you cannot use switches with a MSI package in a GPO. If you don't want to create a MST file you can create a startup script instead. This is a bit of a "cowboy" solution, so for large environments we would recommend System Center Configuration Manager 2007 instead.

Create a **file share** and **copy** the momagent.msi file to that share. Then **create** an **agentdeployment.cmd** file with the following syntax

```
IF EXIST "C:\Program Files\System Center Operations Manager 2007\HealthService.exe"
GOTO end
net use X: \\SRFILE01\agent
X:
CALL momagent.msi /qn USE_SETTINGS_FROM_AD=1
USE_MANUALLY_SPECIFIED_SETTINGS=0
C:
net use X: /DELETE
:end
```

In this example, we have a share named **Agent** on a machine named SRFILE01. This script will check if the agent is present, and if not found, it will install the agent. This script is based on the default installation target directory. Make sure computers that will run this startup script have access to this **Agent** share.

1. Use GPMC to create a new GPO and browse to the following setting
   Computer Settings\Windows Settings\Scripts
2. Right click **Startup** and choose **Properties** from the context menu
3. At the Startup Properties screen, click **Add…** and browse to your file share and select agentdeployment.cmd, click **OK**



4. At the Startup Properties screen, click **OK**

5. Close GPMC. Make sure your target machine is under an OU to which this GPO is linked.

6. When complete, you will need to again approve the manual agent installation as previously illustrated.

Deploy Agent as Part of an Operating System Image

While illustration of the process is beyond the scope of this document, it is also possible to deploy the Operations Manager 2007 Agent as part of an Operating System Image or through Software Distribution features of System Center Configuration Manager 2007.

For more information on System Center Configuration Manager 2007, visit the product homepage at http://www.microsoft.com/sccm

Active Directory Integration in Child and Remote Domains

Since the service connection points used in Active Directory Integration are written to the domain partition of the Active Directory, you will need to run the MOMADAdmin tool in each child domain where you want manually installed agents to retrieve their Management Group settings from the Active Directory.

There are several other factors to bear in mind when configuring AD integration in environments with multiple domains and/or multiple Active Directory forests.

- **Domain functional level must be higher than 'Windows 2000 Mixed'** – Most Active Directory environments today probably meet this requirement, but it is always good to check prior to implementation.

- **RunAs User Account (in each domain)** - RMS performs AD querying and writing with a user account. Domain user rights are sufficient for this account. *When working only with the local or trusted domains, it is optional as the RMS' machine account may be used*. However; using a RunAs Account instead of the RMS' name prevents from having to reconfigure the container objects when the RMS role is moved.

- **Security Group (in each domain)** - Above user account will be made a member of a security group. *For local and trusted domains the existing group, that the OpsMgr administrators are members of, should be used.*

- **LDAP access (RMS to each domain)** - The RMS server needs LDAP access (TCP 389) to at least one DC of each domain. Check if firewalls are blocking traffic to remote domain controllers.

- **DNS resolution (RMS to each domain)** - Optional: If the RMS is able to resolve the DNS namespace of untrusted domains, the configuration doesn't have to rely on IP addresses.

Additional configuration steps

Configuring Active Directory integration for child and/or remote (untrusted) domains requires some additional steps. They have to be repeated for each domain respectively every management server / gateway.

Run MomADAdmin specifying RunAs Account

1. MomADAdmin.exe must be run specifying the RunAs Account name as 3rd parameter. *Do not use the Root Management Server name for remote domains.*

   Example: `MomADAdmin.exe lab "RDOM\OpsMgr Administrators" RDOM\AgtUsr RDOM`

Define RunAs Account

2. From SCOM Operations Console's Administration pane
3. Right-click **Run As Accounts** and choose **Create Run As Account**
4. Run As Account type must be **Windows**

Add Run As Profile

1. From SCOM Operations Console's Administration pane
2. Right-click **Run As Profiles** and choose **Create Run As Profile**
3. Type a name (e.g. AD Based Agent Assignment Account (DOMAIN NAME))
4. Choose '**Default Management Pack'** as destination
5. Add the Run As Accounts created above: *Target must be the Root Management Server*

FYI *It is not normally recommended to save customizations in SCOM to the Default Management Pack. However; RunAs Profiles used for AD integration can only be used when they are defined there.*

Configure Auto Agent Assignment

1. From SCOM Operations Console's Administration pane choose the management server (gateway) located in the remote domain
2. Right click and select **properties**
3. On the Auto Agent Assignment tab, click **Add…**
4. On the domain screen type in the name of your remote domain, remote domain server or its IP address

5. Tick the box **Use a different account to perform agent assignment in the specified domain**
6. From the drop down box, choose the Run As Profile defined above
7. On the inclusion and exclusion screens configure as usual
8. Manually configure failover: Make sure to check only management servers (gateways) located on the same remote domain

Disable RunAs Account Alerts

SCOM will alert about RunAs Accounts failing when remote, untrusted domains are integrated. The Root Management Server attempts to check the RunAs Account validity and fails since that account does not exist in its own domain. In order to disable these alerts:

Option A - Disable the monitors completely

If you disable the RunAs Account monitors completely, you will not receive alerts for other (local) RunAs Accounts at the Root Management Server.

1. From SCOM Operations Console, choose Authoring
2. Click Monitors and change the scope to Root Management Server
3. Look for these monitors and disable using overrides for all objects of type: Root Management Server:
    - RunAs Account Monitoring Check
    - RunAs Successful Logon Check

Option B – Replace the monitors

This sample management pack disables the original monitors and replaces them with alerts that allow filtering on the RunAs Accounts' names. That way only the RunAs Accounts for remote domains are excluded from monitoring.

1. Import the management pack
2. From SCOM Operations Console, choose **Authoring**
3. Click **Monitors** and change the scope to **Root Management Server**
4. Look for these monitors and open properties
    a. RunAs Account Monitoring Check (Replaced)
    b. RunAs Successful Logon Check (Replaced)
5. Choose the **Expression (Unhealthy Event)** tab
6. Modify, add or delete the EventDescription expressions as required

Agents that Cannot Participate in Active Directory Integration

There are some agents that cannot participate in Active Directory integration.

- **Push-installed Agents** – Because the management group settings are defined during the push-install process, agents installed with this method will not read and use settings from the Active Directory. They might however pick up settings for an additional management group configured using Active Directory integration. They will then report to both management groups.

- **Agents on Domain Controllers** - When agents query Active Directory to identify their Management Group settings, the settings revealed are determined by the SCPs for which the agent has read access. Since the Health Service runs in the LocalSystem context, on a domain controller the service has read access to every SCP. This could be a big issue in the case where multiple Management Groups are published in the Active Directory, as the DC's Health Service would try to participate in every AD-integrated management group.

  Although it is not possible that an agent on a domain controller participates in Active Directory integration, warning event 211 is written to the Operations Manager event log every time the health service is started. This event can be ignored.

Troubleshooting AD Integration

There are several events logged in the situations where AD integration is not functioning as expected. These events are generally descriptive of the problem.

Event 20064 on agent (multiple primary relationships)

- LDAP queries overlap
- Improper permissions on OperationsManager container
  → Agents connect to wrong MS - Event 20064 (Manageability Team Blog)

Event 20070 on agent (agent not authorized)

- Agent was not acknowledged (see pending management)
- MS or GW does not have read access to computer account's container
  → Agents unable to communicate in remote domains (Manageability Team Blog)

Event 21016 on agent (no failover)

- Specify valid failover configuration in AD assignment rule (do not use automatic setting)
- Check that **[MSName2]_PrimarySG_[number]** is a member of [**MSName1]_SecondarySG_[number]** security group and vice versa
  - Appears in untrusted domain set up using gateways (scheduled to be fixed in R2 at the time of this writing)
  - Workaround: Add the Primary security groups manually to the secondary security groups. Repeat every time groups are recreated

Event 21034 on agent (no configured parents)

- No LDAP query returns the account of the computer
- Computer has only recently been joined to AD and AD assignment rule has not run since
  → Try again an hour later

Registry Keys Related to AD Integration Behavior

There are a few registry keys on the agent that can provide some measure of control or insight into agent configuration and behaviour with regard to AD integration.

Enable AD Integration Key
HKLM\SYSTEM\CurrentControlSet\Services\HealthService\Parameters\ConnectorManager
EnableADIntegration (DWord)

Comments: This key should not normally be changed. As mentioned earlier, the warning Event 211 on DCs can be safely ignored. If necessary, set this value to 0 to prevent a push installed agent from reading configuration from AD for an additional management group. This may be desirable to prevent high security / mission critical servers from reporting to additional management groups installed in the future.

AD Poll Interval
HKLM\SYSTEM\CurrentControlSet\Services\HealthService\Parameters\ConnectorManager
ADPollIntervalMinutes (DWord)

Comments: Agent polls AD every 60 minutes by default. If you absolutely must, you can add this key to change the polling frequency.

Is an agent using configuration retrieved from AD?
HKLM\SYSTEM\CurrentControlSet\Services\HealthService\Parameters\Management Groups\[MGName]
IsSourcedFromAD (DWord)

Comments: If above key is present then an agent has read the configuration for that management group from AD.

LDAP Query Syntax and Samples

Sample Queries

Below are some sample queries to provide numerous options for filtering LDAP query results when configuring auto assignment.

Limit the query to computer accounts

- (objectCategory=computer)

OR

- (sAMAccountType=805306369)

Excludes Domain Controllers

- (!(primaryGroupID=516))

Excludes OpsMgr Management Servers and Gateways

- (!(servicePrincipalName=MSOMHSvc/*))

Direct members of a security group

- (memberOf:=CN=Admin,OU=Security,DC=DOM,DC=NT)

Resolves nested security groups (requires at least Windows 2003 SP2)

- (memberOf:1.2.840.113556.1.4.1941:=CN=Admin,OU=Security,DC=DOM,DC=NT)

Returns odd servers if their netbios names end with a number (e.g. AnySrv101)

- (|(name=*1)(name=*3)(name=*5)(name=*7)(name=*9))

Combination sample

- (&(objectCategory=computer)(!(primaryGroupID=516))(!(servicePrincipalName=MSOMHSvc/*))(|(name=*1)(name=*3)(name=*5)(name=*7)(name=*9)))

OU Membership queries

Assigning agents based on their Organizational Unit (OU) membership is possible but requires additional work. LDAP does not support using wildcard matching on an object's distinguishedName (DN) property from which the OU location can be derived. A direct query like **the following will not work**:

(&(objectCategory=computer)(distinguishedName=*,OU=OrgUnit,DC=Domain,DC=info))

Instead either one of the following two solutions can be used.

Option A – Use security groups per OU

Create a scheduled task that executes a command to query an OU and add the computers in there to a security group. This can be accomplished with VBScript, Powershell, or even the AD command line tools

For example, you can use the DSQuery and DSMod tools to do this:

```
dsquery computer "OU=myOU,DC=myDomain,DC=com" | dsmod group
"CN=MgmtServer1AgentGroup,CN=Users,DC=myDomain,DC=com" -chmbr
```

(This adds all computers found in myOU to the security group MgmtServer1AgentGroup)

Then, in agent assignment & failover wizard, you can then include all members in that security group using a LDAP filter like:

```
(&(sAMAccountType=805306369)(memberof=CN=MgmtServer1AgentGroup,CN=Users,DC
=myDomain,DC=com))
```

Option B – Use otherwise empty attribute

If creating additional security groups is not an option, the OU location can be copied into an unused property. Good candidates are 'comment' or 'info'. They are usually empty and do support wildcard matching in LDAP queries.

The AD Integration Demo MP includes a script which does that automatically. It uses the 'info' attribute but that can be changed using an override. The LDAP filter will then look like this:

```
(&(sAMAccountType=805306369)(!(primaryGroupID=516))(info=*,OU=SubUnit,OU=OrgUni
t,DC=Domain,DC=info))
```

LDAP Syntax

The tables below contain common Active Directory attributes for computer accounts that can be used in LDAP queries in AD integration, as well as an explanation of common LDAP operators.

Common Active Directory Attributes for LDAP Queries

Computer Account Attribute	Remark
description	Computer description (in AD)
distinguishedName	DN: OU location of the computer account can be read from here. ***No wildcard matching possible!***
dNSHostName	FQDN
location	Location field
memberOf	Groups the computer account is a member of. ***No wildcard matching possible!***
name	Netbios computer name
operatingSystem	e.g. Windows Server 2003
operatingSystemServicePack	e.g. Service Pack 1
operatingSystemVersion	e.g. 5.2 (3790)
primaryGroupID	515: Computers 516: Domain Controllers
sAMAccountName	Computer account name ([name]$)
sAMAccountType	always 805306369 (computer account)
servicePrincipalName	list of registered SPNs

LDAP Operators

Operator	Description
|	OR
&	AND
!	NOT
=	Equals
~=	Approx. equals
<=	Less than or equal
>=	More than or equal

LDAP Escape Sequences

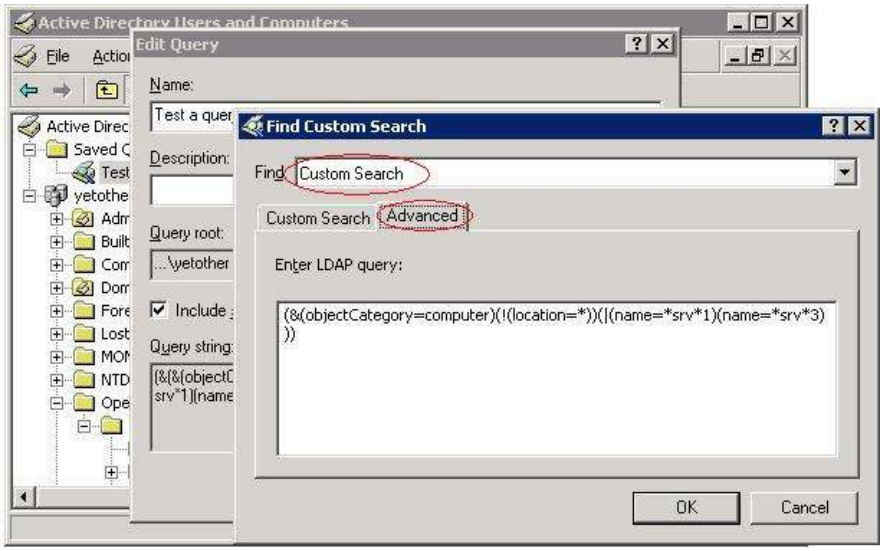
ASCII character	Escape sequence
*	\2a
(	\28
)	\29
\	\5c
NUL	\00

Any of above characters must be replaced by the escape sequence if used in matching values

Testing LDAP filters

The 'Active Directory Users and Computers' snap-in does offer support for testing LDAP queries.

1. Open '**Active Directory Users and Computers'**
2. Click on the '**Saved Queries'** folder
3. From the Actions menu choose **New → Query**
4. Enter a name and click **'Define Query'**
5. From the Find drop down list choose **'Custom Search'** and click on **the 'Advanced'** pane
6. Type in your LDAP filter expression



Additional Resources

How to use Group Policy to remotely install software in Windows Server 2003
http://support.microsoft.com/kb/816102

Group Policy Management Console with Service Pack 1
http://www.microsoft.com/downloads/details.aspx?FamilyID=0A6D4C24-8CBD-4B35-9272-DD3CBFC81887&displaylang=en

Creating an LDAP Query Filter
http://msdn2.microsoft.com/en-us/library/ms675768.aspx

Feedback

I hope you find this article helpful. Your feedback is always welcome and appreciated at administrator[AT]systemcenterforum.org