

## Windows Server® 2008 Active Directory® Resource Kit

*Stan Reimer, Mike Mulcare,  
Conan Kezema, Byron Wright  
w MS AD Team*

**PREVIEW CONTENT** This excerpt contains uncorrected manuscript from an upcoming Microsoft Press title, for early preview, and is subject to change prior to release. This excerpt is from *Windows Server® 2008 Active Directory® Resource Kit* from Microsoft Press (ISBN 978-0-7356-2515-0, copyright 2008 Mike Mulcare (Content); Stan Reimer (Content), all rights reserved), and is provided without any express, statutory, or implied warranties

To learn more about this book, visit Microsoft Learning at  
<http://www.microsoft.com/MSPress/books/9552.aspx>

**Microsoft®**  
Press

978-0-7356-2515-0

© 2008 Mike Mulcare (Content); Stan Reimer (Content). All rights reserved.

# Table of Contents

## **9 Delegating the Administration of Active Directory**

### **Active Directory Object Permissions**

- Standard Permissions
- Special Permissions
- Permissions Inheritance
- Effective Permissions
- Ownership of Active Directory Objects

### **Auditing the Use of Administrative Permissions**

### **Delegating Administrative Tasks**

### **Customized Tools for Delegated Administration**

- Customizing the Microsoft Management Console

### **Planning for the Delegation of Administration**

## Chapter 9

# Delegating the Administration of Active Directory Domain Services

Active Directory Domain Services (AD DS) is typically deployed as a common directory service shared between various business divisions within an organization. Using a common directory service helps reduce the costs associated with maintaining the infrastructure, but introduces a number of other considerations such as:

- How to manage users and resources independently between divisions when decentralized administration is required.
- Ensuring that administrators or users can only perform permitted tasks within their own business division.
- Ensuring that specific objects or information stored within the directory is only available to administrators with the appropriate permissions.

These considerations can be addressed by a thorough understanding of how to delegate administration tasks. Delegation involves a higher-level administrator granting permissions to other users to perform specific administrative tasks within the Active Directory structure. The Active Directory structure provides a hierarchical view of the directory service; first at the site and domain level, and then at the organizational unit (OU) level within a domain. This hierarchy provides powerful options for managing permissions and delegating administrative tasks at various levels throughout the logical infrastructure.

This chapter describes administrative delegation, starting with a discussion on the various types of tasks that might be delegated within an enterprise. This chapter then describes object access, the types of permissions that can be assigned to objects residing within the directory, and how to use these permissions for delegation of administration. Finally this chapter provides information on auditing changes to objects residing within AD DS.

## Active Directory Administration Tasks

Active Directory administration tasks typically fall into one of two categories; data management or service management. Data management tasks relate to the management of content that is stored within the Active Directory database. Service management tasks relate to the management of all aspects that are required to ensure a reliable and efficient delivery of the directory service throughout the enterprise.

Table 9-1 describes some of the tasks that are related to each of these categories:

Table 9-1 Active Directory Administration

Category	Tasks
Data management	<ul style="list-style-type: none"> <li>• <b>Account management</b> - includes creating, maintaining, and removing user accounts.</li> <li>• <b>Security group management</b> - includes creating security groups, provisioning security groups to grant access to network resources, managing memberships of security groups, and removing security groups.</li> <li>• <b>Resource management</b> - includes all aspects of managing network resources such as end-user workstations, servers, and resources hosted on servers such as file shares or applications.</li> <li>• <b>Group Policy management</b> - includes all aspects of creating, assigning, and removing Group Policy objects within the Active Directory structure.</li> <li>• <b>Application-specific data management</b> - includes all aspects of managing Active Directory-integrated or -enabled applications such as Microsoft Exchange Server.</li> </ul>
Service management	<ul style="list-style-type: none"> <li>• <b>Installation and trust management</b> - includes aspects such as the creation and deletion of domains, the deployment of domain controllers, and the configuration of appropriate Active Directory functional levels.</li> <li>• <b>Domain controller and directory database management</b> - includes aspects related to the management of domain controller hardware, database maintenance, and the application of service packs and security updates.</li> <li>• <b>Schema management</b> - includes the extension or modification of the schema to support the deployment of Active Directory-enabled applications.</li> <li>• <b>Operations master roles management</b> - includes tasks that ensure the proper assignment and configuration of operations master roles.</li> </ul>

- 
- **Backup and restore management** - includes all tasks related to performing regular backups of the directory database and restore procedures when required.
  - **Replication management** - includes all tasks related to the creation, maintenance, and monitoring of the replication topology.
  - **Security policy management** - includes all tasks related to the management of the default domain controller security policy and managing the password, account lockout, and Kerberos account policies.
- 

[Note] For more information about the tasks related to data management and service management, refer to Best Practices for Delegating Active Directory Administration found at <http://www.microsoft.com/technet/prodtechnol/windowsserver2003/technologies/directory/activedirectory/actdid1.mspx>

Delegating data and service management tasks within an organization requires an understanding of the administrative needs of all business units. This understanding ensures the most effective delegation model used to provide a more effective, efficient, and secure networking environment. To deploy the delegation model, you need to understand Active Directory object permissions, delegation methods, and auditing. These concepts are discussed in the next few topics.

## Accessing Active Directory Objects

To effectively delegate administration tasks, you need to know how Active Directory controls access to objects stored within the directory service. Access control involves the following:

- Credentials of the security principle attempting to perform the task or access the resource.
- Authorization data used to protect the resource or authorize the task being performed.
- An access check that compares the credentials against the authorization data to determine whether the security principle is permitted to access the resource or perform the task.

When a user logs on to an AD DS domain, authentication takes place and the user receives an access token. An access token includes the security identifier (SID) for the user account, SIDs for each security group to which the user is a member of, and a list of privileges held by the user and the user's security groups. The access token helps to provide the security context and credentials needed to manage network resources, perform administrative tasks, or access objects residing in Active Directory.

Security is applied to a network resource or an Active Directory object by authorization data that is stored in the *Security Descriptor* of each object. The Security Descriptor consists of the following components:

**Object owner**

- The SID for the current owner of the object. The owner is typically the creator of the object or a security principle that has taken over ownership of an object.

**Primary group**

- The SID for current owner's primary group. This information is only used by the Portable Operating System Interface for UNIX (POSIX) subsystem.

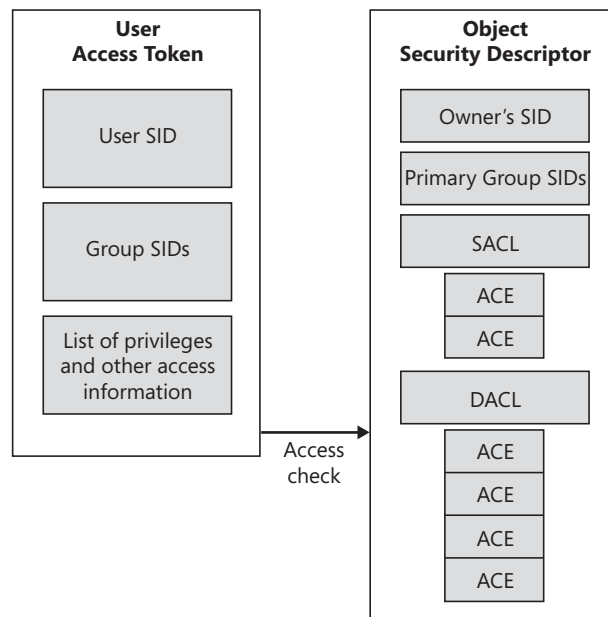
**Discretionary access control list (DACL)**

- A list of access control entries (ACEs) that define the permissions each security principle has to an object. Each security principal that is added to the access control list obtains a set of permissions that specify the extent to which that user or group can manipulate the object. If a user does not appear in an ACE, either individually or as a member of a group, that user has no access to the object.

**System access control list (SACL)**

- Defines the audit setting on an object including which security principle is to be audited and the operations that are to be audited.

Figure 9-1 illustrates the architecture of a user's access token and an object's security descriptor. When a user tries to access a network resource or an Active Directory object, an access check is performed and each ACE is examined until a User or Group SID match is found. Access is then determined by the permissions configured on the ACE.



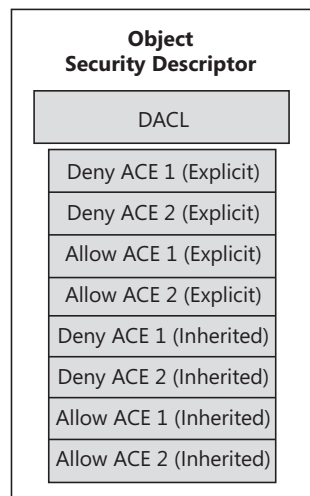
**Figure 9-1** Access check between a user's access token and an object's security descriptor.

## Evaluating Deny and Allow ACEs in a DACL

ACEs are listed within a DACL in a specific order, which has a direct affect on the outcome of the access check. During an access check, ACEs are evaluated in sequence. The evaluation sequence is listed as follows:

- ACEs that have been explicitly assigned are evaluated before inherited ACEs.
- For each set of explicit or inherited ACEs, Deny ACEs are always evaluated before Allow ACEs.

Figure 9-2 illustrates how Allow and Deny permissions are evaluated for both explicit and inherited ACEs.



**Figure 9-2** Evaluating Allow and Deny ACEs.

## Active Directory Object Permissions

Every object in Active Directory has an ACL, which means that you can modify the permissions on that object. This includes objects visible through the Active Directory Users and Computers administrative console as well as objects visible through the Active Directory Sites and Services administrative console, ADSI Edit, or Ldp.exe. The most common tool used to modify Active Directory object access is Active Directory Users and Computers. However, each of the above mentioned tools can be used to perform the common task of managing object access within the directory service.

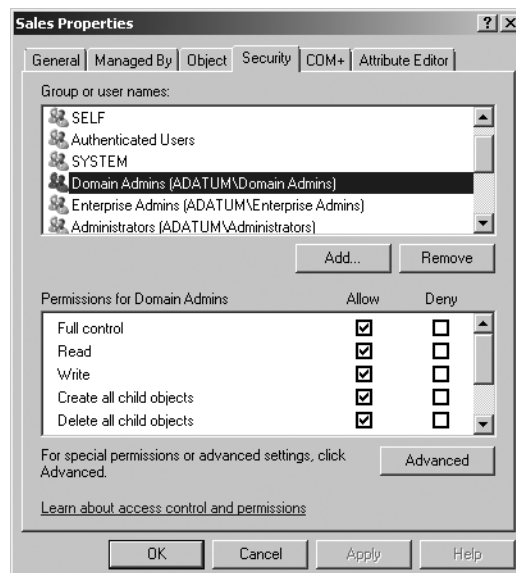
Access control permissions on an Active Directory object are separated into two categories: *standard permissions* and *special permissions*. Special permissions are granular options that can be applied to an object. A standard permission is made up of a group of special permissions to allow or deny a specific function. For example, the Read standard permission is made up of the Read permissions, List contents, and Read all properties special permission entries.

## Standard Permissions

To view the standard permissions for any Active Directory object in the domain directory partition, access the Security page for that object's Properties sheet in the Active Directory Users and Computers administrative console.

[Note] If the Security page is not visible, select Advanced Features on the View menu, then re-select the object and open its Properties sheet.

The Security page displays the group or user names that are assigned permissions to the object. As you select a group or user entry, the associated allow or deny permissions for that entry are shown. Figure 9-3 illustrates the permissions for the Domain Admins group on the Sales organizational unit. Notice that, by default, the Allow box is checked for each permission to provide the Domain Admins group full control over the Sales OU.



**Figure 9-3** Viewing the Security page on an Organizational Unit object.

Depending upon the type of object being secured, you will notice that different permissions may be visible on the security page. For example the following standard permissions are common with all objects:

- Full control
- Read
- Write
- Create all child objects
- Delete all child objects

Some Active Directory objects also have standard permissions that are applied to grouped sets of properties. For example, a user object has several read and write property sets such as General Information, Personal Information, Phone and mail options, and Web



Information. Each of these property sets refers to a set of object attributes, so granting access to a single property set provides access to a set of attributes. For example, the Personal Information property set includes attributes such as *homePhone*, *homePostalAddress*, *streetAddress*, and so on. Using the property sets to assign access to groups of attributes simplifies the process of assigning permissions without having to modify at the granular attribute level.

[Note] The Active Directory schema defines which attributes are part of each property set by using the rightsGuid value for the property category (in the Configuration directory partition) and the attributeSecurityGUID for the schema object. For example, the rightsGuid value for cn=Personal-Information, cn=Extended-Rights, cn=configuration, dc=forestname is equivalent to the attributeSecurityGUID for cn=Telephone-Number, cn=Schema, cn=Configuration, dc=forestname. This means that the telephone number is included in the Personal Information property set.

In addition to the standard permissions, the Security page may also show extended rights related to the object being secured. Depending upon the object, these rights include options such as Allowed to authenticate, Generate resultant set of policy, Receive as, Send as, Send to, Change password, and reset Password.

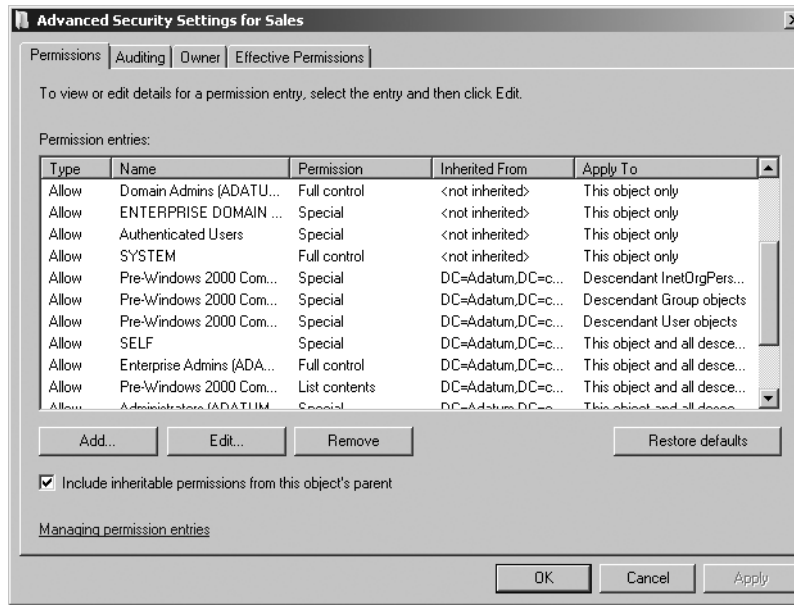
## Special Permissions

One of the entries in the permissions list on the Security page is Special permissions. In addition to being able to grant standard permissions, you can also grant special permissions to Active Directory objects.

[Note] You can determine if special permissions are applied to an object by viewing the Allow or Deny check boxes located next to the Special permissions entry. If a check mark is visible, special permissions have been assigned.

As mentioned previously, special permissions are much more granular and specific than standard permissions. To simplify management, you would typically use standard permissions on an object; however, there may be specific needs that require you to modify the special permission entries.

To get access to special permissions, click Advanced on the Security page and then ensure that the Permissions page is selected. Figure 9-4 shows the interface. Table 9-2 explains the options available on the Permissions page.



**Figure 9-4** Viewing the Advanced Security Settings for an object.

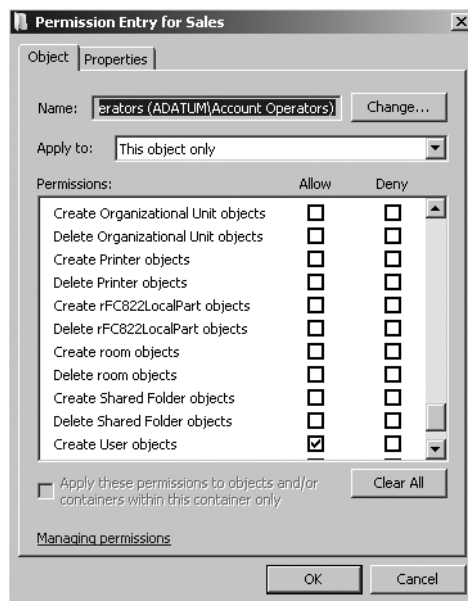
**Table 9-2. Special Permissions Configuration**

Option	Explanation
Type	This value is set to either Allow or Deny. Normally, the interface sorts the permissions so that all Deny permissions are listed first, but the sort order can be changed by clicking any column header. Regardless of the order of appearance in this column, the Deny permissions are always evaluated first.
Name	This is the name of the security principal that each ACE applies to.
Permission	This column lists the level of permission granted for the security principal. Levels of permission can be standard rights, such as Full Control; special permissions such as Create/Delete User Objects; or just Special. The types of permissions available depend on the type of object and how granular you have configured the permission entry.
Inherited From	This column lists the location where this permission is set and if the permission is inherited from a parent container.
Apply To	This column specifies the depth to which this permission applies. It has a variety of settings, including This Object Only, This Object And All Child Objects, or Only Child Objects.
Include inheritable permissions from this object's parent	This option allows you to specify whether parent permission entries are to be applied to the object.
Add/Edit/Remove buttons	These buttons allow to your add new ACEs, Remove existing ACEs, or edit a specific ACE to provide more granular permission settings.

[Note] The Restore defaults button on the Permissions page resets the permissions on the object to the default permission settings.

In many cases, the same security principals may be listed in multiple ACEs. For example, the Authenticated Users group has multiple Allow entries for Read permissions, Read general information, Read personal information, Read web information, and Read public information in separate ACEs. This happens whenever you specify a combination of permissions that cannot be stored within a single ACE.

If you add or edit the permissions granted to a security principal, you are provided two different options for applying permissions. Figure 9-5 shows the first option, which is applying permissions to the object.



**Figure 9-5** Assigning special permissions to Active Directory objects.

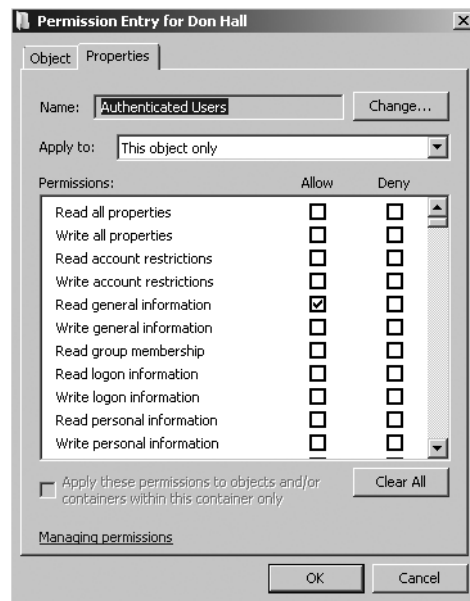
The Object tab is used to apply permissions to various object scopes such as:

- **This object only.** Permissions only apply to the object being secured or modified.
- **This object and all descendant objects.** Permissions will apply to both the object being secured and all child objects within the object.
- **All descendant objects.** Permissions will only apply to child objects within the object being modified.
- **Individual descendant objects.** Windows Server 2008 provides a large selection of individual descendant objects that can be granularly secured. For example, if you are assigning permissions at the OU level, you may choose to only apply permissions to computer objects within the Sales OU. These options provide the capability to delegate permissions at a granular object level.

The second option for applying permissions is to control access to the object properties. Figure 9-6 shows the interface.

The Properties page is used to apply permissions for the security principle listed in the Name field to the individual properties for the object. For example, if you are applying permissions to a user object, you are given the option of assigning Read and Write

permissions to each attribute available on the object class, such as general information, group membership, personal information, and so on.



**Figure 9-6** Configuring an object's property permissions.

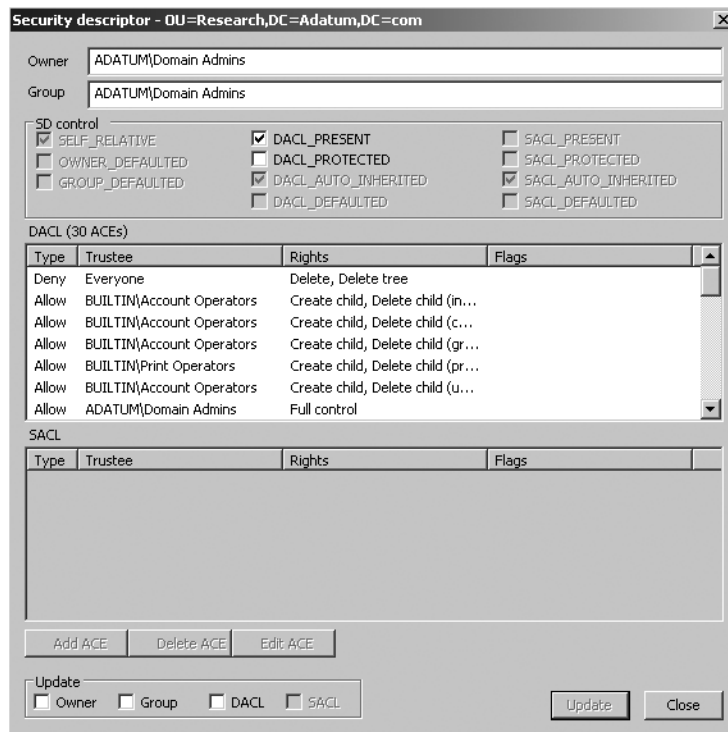
## Viewing the ACE using Ldp.exe

Ldp.exe is a graphical user interface (GUI) tool used to perform operations such as connect, bind, search, modify, add, or delete against any LDAP-compatible directory service. LDP can be used to view advanced Active Directory metadata such as security descriptors and replication metadata.

To view the ACL using Ldp.exe:

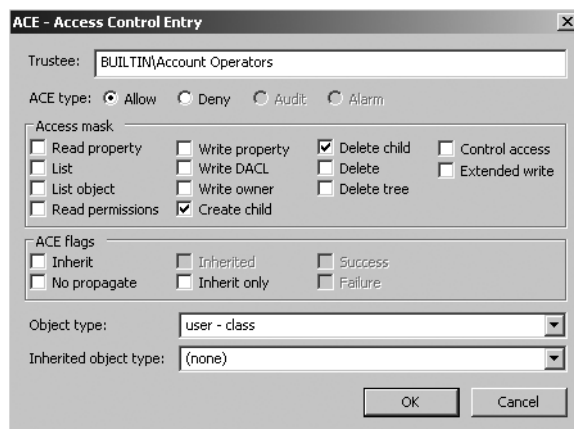
1. Open the Run dialog box and type **ldp**.
2. Click the **Connection** menu and then click **Connect**.  
If you leave the server box empty, the server will connect to the local computer. You can also type in the server name.
3. Once you are connected to the server, click the **Connection** menu and then click **Bind**. If you are not logged in with a user account that has administrative rights, type in alternate credentials. Otherwise, leave the logon information blank.
4. After binding to the domain, click the **View** menu and then click **Tree**.
5. To view the entire domain, click **OK**. The domain OU structure will be listed in the left pane.
6. To view the ACL for any object, locate the object in the tree view in the left pane. Right-click the object, point to **Advanced**, and then click **Security Descriptor**.

As shown in Figure 9-7, a number of advanced options are available such as modifying DACL and SACL rights and modifying the security descriptor controls such as DACL and SACL protection.



**Figure 9-7** Using Ldp.exe to modify the security descriptor.

When you add or edit an ACE using Ldp.exe, you are able to modify specific permissions and ACE flags on various object types and specify object inheritance. Figure 9-8 shows an illustration of the ACE editor provided with Ldp.exe.



**Figure 9-8** Modifying an ACE using Ldp.exe.

## Permissions Inheritance

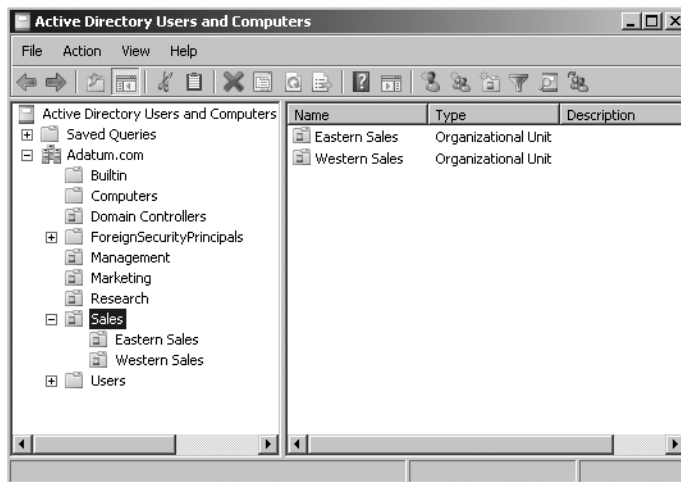
AD DS uses a static permissions inheritance model. That is, when permissions are changed on a container object in the Active Directory structure, that permission is calculated and applied to the security descriptor for all objects in that container. Consequently, if permissions are changed higher in the Active Directory structure and these permissions are applied to all child objects, calculating the new ACL for each object can be a processor-intensive process. However, this initial effort means that the permissions do not need to be recalculated when a user or process tries to access the object.

There are two primary methods that are used to control inheritance of permissions. These methods include:

- **Configuring inheritable permissions on the object.** By default, when an object is created in Active Directory, inheritable permissions are included from the object's parent. You can determine if a permission entry is inherited by the shaded check box on the Security page, or from the Inherited From column of the Advanced Security Settings box.
- **Configuring the scope of how permissions are applied.** As described previously, another way to control inheritance is to specify how permissions apply to descendant objects when security is applied to an object. By default when a new group or user name is manually added to the ACE, the entry has permissions that apply to *this object only*. To force inheritance to child object, you need to modify the scope to apply to descendant objects in addition to the current object.

[Note] If you use the Delegation of Control Wizard, inheritance will automatically be set to This object and all descendant objects. More information about the Delegation of Control Wizard is provided in the Delegating Administrative Tasks section later in this chapter.

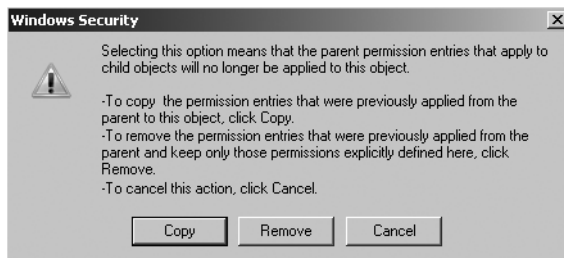
If you have designed your OU structure with the goal of delegated administration, you will have created an OU structure where top-level administrators that require permissions to all Active Directory objects are granted permissions high in the hierarchy with delegated permissions to all descendant objects. As you move further down the hierarchy, you may be delegating permissions to other administrators who should only have control over a smaller part of the domain. For example, Figure 9-9 shows the Sales OU. Within the Sales OU are two child OUs called Eastern Sales and Western Sales. The manager that is in charge of the entire Sales division may be delegated permissions to the entire Sales OU object and all descendent objects, whereas the Eastern Sales or Western Sales managers may be delegated permissions only to their own specific OU only.



**Figure 9-9** Delegating management of the Sales OU.

In some cases, however, you may want to block higher-level administrators from having any administrative permissions to a specific child OU. For example, you may have created a child OU for a branch office in your company, and you may assign a local administrative group full control of the OU. You may, however, not want those local administrators to have access to any executive user accounts in the OU. To accomplish this, you can create an Executives OU within the branch office OU and then remove the option to include inheritable permissions from the object's parent. This in effect, blocks permissions inheritance at the Executives OU level.

To block the inheritance of permissions on an Active Directory object, access the Advanced Security Settings dialog box for the object (shown in Figure 9-4). Then clear the option to Include inheritable permissions from this object's parent. When you clear this option, you are presented with the choice to copy the existing permissions or remove all permissions before explicitly assigning new permissions as shown in Figure 9-10.



**Figure 9-10** Selecting the option to copy or remove permissions when blocking permissions inheritance.

Blocking inheritance has the following implications:

- The permissions are blocked for the object and any descendant objects. This means that you cannot block the permissions inheritance at a container level and then reapply the inheritance from a higher container at a lower level.
- Even if you decide to copy the permissions before modification, permissions inheritance begins where you block the permissions. If you modify the permissions at a higher level, the permissions will not be inherited past the blocked permissions.

- You cannot be selective about what permissions are blocked. When you block permissions, all inherited permissions are blocked. Permissions that have been explicitly assigned to the object or child objects are not blocked.

[Note] One of the possible concerns with blocking inherited permissions is that you might create an orphaned object where no one has any permissions. For example, you can create an OU, block all permissions inheritance to that OU, and assign the permissions to only one administrative group. You can even remove the Domain Admins group from the ACL of the OU so that the Domain Admins does not have any permissions under normal circumstances. If that administrative group gets deleted, the OU would have no group with administrative control. In this case, the Domain Admins group would have to take ownership of the object and reassign permissions.

## Effective Permissions

As discussed so far in this chapter, a user can obtain permissions to a specific object in Active Directory in several ways. These include:

- The user account may be granted explicit permissions to an object.
- One or more groups that the user belongs to may be granted explicit permissions to an object.
- The user account or one or more groups that the user belongs to may be given permissions at a container-object level and permissions inherited by lower-level objects.

All of these permissions are cumulative, that is, the user is granted the highest level of permissions from any of these configurations. For example, if a user is explicitly given Read permission to an object, the user belongs to a group that is explicitly given Modify permissions, and the user belongs to a group that is given Full Control at the container level, the user will have Full Control. When a user attempts to access an object, the security subsystem examines all of the ACEs that are attached to the object. All of the ACEs that apply to the security principal (based on user account or group SIDs) are evaluated and the highest level of permission is set. However, in addition to ACEs that grant permissions, Active Directory also supports Deny permissions. Deny permissions can be applied at two levels.

- The user object or one or more of the groups that the user belongs to may be explicitly denied permission to an object.
- The user object or one or more groups that the user belongs to may be denied permissions at a container level, and this denial of permission may be inherited to lower-level objects.

Deny permissions almost always override Allow permissions. For example, if a user is a member of a group that is given Modify permissions to an Active Directory object, and the user is explicitly denied Modify permissions to the object, the user will not be able to modify the object. This is because the ACEs that deny permissions are evaluated before the ACEs that allow permissions. If one of the ACEs denies permission to the security principal, no other ACEs are evaluated for the object.



The one situation where Allow permissions do override Deny permissions is when the Deny permissions are inherited and the Allow permissions are explicitly assigned. For example, you can deny a user the permission to modify any user accounts in a container. But, if you explicitly allow Modify permissions to an object within the container, the user account will have Modify permissions on that object.

---

### **Deny Permissions: Use Carefully**

Using the Deny option to deny permissions can make your Active Directory security design very difficult to manage. There are a number of different scenarios where you may think about using the Deny permission. One is that you may want to use the Deny option to remove some permissions that are being inherited. For example, you may grant Modify permissions at a container level, but may want to change that to Read-Only further down the hierarchy. In this case you could deny the Write permission on any objects or properties further down the hierarchy.

Another scenario where you may think of using the Deny option is when you want to create a container that requires higher security. For example, you may have a container for all of the executives, and you may want to make sure that a normal user cannot read the executive account properties. You may choose to deny Read permissions on the container using the Domain Users group. Unfortunately, this denies everyone the right to read the directory objects, including all administrators. Because of the complications that can result from using the Deny option, you should use it with care.

In most cases, rather than denying permissions you can just ensure that a user or group has not been given permissions. If a user has not been granted any permissions and is not a member of any group that has been granted permissions, the user will not have any access and will be implicitly denied. You do not need to explicitly apply the Deny permission to prevent users from accessing objects in Active Directory.

One of the few scenarios in which it can be beneficial to use the Deny option is if you have a case where a group should be given permissions but one or more users in the same group should have a lower level of permissions. For example, you may have a group called Account Admins that is responsible for managing all user accounts in the domain. Some members of this group may be temporary employees who need to be able to manage all user accounts in the domain, but should not be able to modify any properties on executive accounts. In this case, you could assign the Account Admins group permission to manage all user accounts in the domain. Next, create an OU for the executive accounts, and create a group for the temporary members of the Account Admins group. Then deny the temporary users the right to modify any user accounts in the Executive OU.

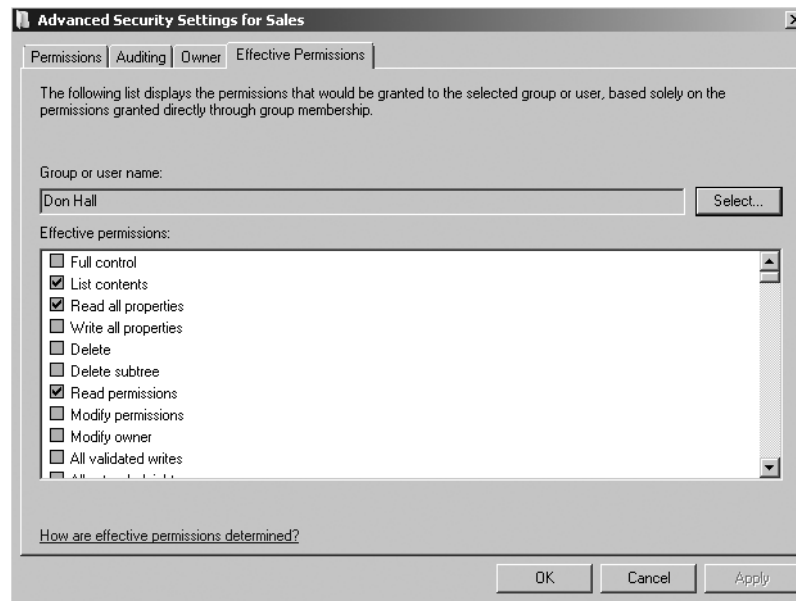
---

As you can see, configuring security on Active Directory objects can involve managing a large number of interrelated variables. Many companies may start out with a fairly simple security design where a small group of administrators are given all the permissions in Active Directory. Most of the time, the initial Active Directory security configuration is clearly documented. However, as time goes by, this simple initial configuration often becomes much messier. Sometimes another group of administrators is given a set of permissions for a specific task and for a specific period of time. Granting the permissions is easy to do, but often the permissions are never removed. Often these security modifications made after the initial deployment are also not clearly documented.

For any Active Directory structure that has been deployed for some time, the current security configuration is likely more complex than was initially designed. Sometimes this results in users having more permissions than they should have. Fortunately, Windows Server 2008 provides a tool that can be used to easily determine the effective permissions a security principal has to any object in Active Directory.

To determine the effective permissions that a security principal has on an Active Directory object, access that object's properties through the appropriate Active Directory administrative tool. Click the Security page, click Advanced, and then click the Effective Permissions page. To determine the effective permissions for a specific user or group account, click Select and then search for the user or group name. After you have selected the name, click OK. The Effective Permissions page displays all of the permissions the security principal has to the Active Directory object. Figure 9-11 shows the interface for the Active Directory Users and Computers administrative tool. Notice that the Effective Permissions page for the Sales OU displays the overall permissions assigned to the Don Hall user object.

[Note] This tool has some limitations that may affect the effective permissions displayed. The tool determines the effective permissions based on inherited and explicitly defined permissions for the user account and the user's group memberships. However, the user may also get some permissions based on how the user logs on and connects to the object. For example, in Windows Server 2008, you can assign permissions to the Interactive group (that is, anyone logged on to the computer) or the Network Login group (that is, anyone accessing the information across the network). This Active Directory administrative tool cannot determine the permissions granted to a user based on these types of groups. Also, the tool can only determine permissions by using the permissions of the person running the tool. For example, if the user running the tool does not have permission to read the membership of some of the groups that the evaluated user object belongs to, the tool will not be able to determine the permissions accurately.

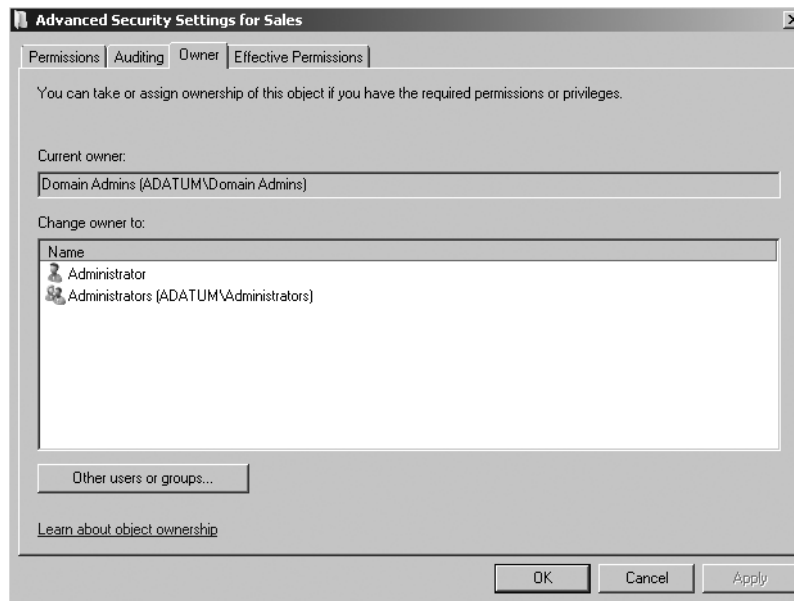


**Figure 9-11** Displaying the effective permissions for an Active Directory object.

## Ownership of Active Directory Objects

Every object in Active Directory has an owner. By default, the user that created an object is the owner. The owner of an object has the right to modify permissions on the object, which means that, even if the owner does not have full control of an object, the owner can always modify the permissions on the object. In most cases, the owner of an object is a specific user account rather than a group account. One exception to this is when an object is created by a member of the Domain Admins group; the ownership of the object is then assigned to the Domain Admins group. If the owner of the object is a member of the local Administrators group but not a part of the Domain Admins group, the ownership of the object is assigned to the Administrators group.

To determine who the owner of an Active Directory object is, access that object's properties using the appropriate Active Directory administrative tool. Select the Security page, click Advanced, and then select the Owner page. Figure 9-12 shows the interface for the Active Directory Users and Computers administrative tool.



**Figure 9-12** Viewing the ownership of an Active Directory object.

If you have the Modify owner permission to the object, you can use this interface to modify the owner of the object. You can choose either to take ownership for your own account or to assign the ownership to another user or group. This last option is unique in Windows Server 2003 and Windows Server 2008 Active Directory. In Microsoft Windows 2000 Active Directory, you could only take ownership of an object; you could not assign the ownership to another security principal.

---

## Administrative Privileges

The administrative permissions discussed so far have to do with specific permissions on Active Directory objects and define what actions the administrator can perform on those objects. In addition to these permissions, a user may also be able to perform some tasks in Active Directory because of the privileges assigned to him or her. The permissions discussed so far are based on the ACLs that are attached to each Active Directory object. User privileges are different because user privileges are applied to user accounts. User privileges are something that the user has because of who he or she is, not because he or she has permission to modify a particular Active Directory object. For example, there are two ways that you can give a user or group the right to add workstations to the domain. One option is to give the user or group permission to Create Computer Objects either at an OU level or at the Computers container level. This allows the user to add as many workstations as needed to the domain in the specified container.

Another way to allow a user to add workstations to the domain is to give him or her the privilege *add workstations to domain*. This privilege is a part of the Default Domain Controllers Policy. Any user who has this privilege can add up to ten workstations to the domain. By default the Authenticated Users group is granted this permission.

---

## Delegating Administrative Tasks

This chapter has thus far discussed how to ensure the security of Active Directory objects. This has been in preparation for this section; which applies the security options to delegate administrative tasks. Because every object in Active Directory has an ACL, you can control administrative access down to any property on any object. This means that you can grant other Active Directory administrators very precise permissions so that they can perform only the tasks they need to do.

While you can get extremely specific about delegating administrative permissions, you should maintain a balance between keeping things as simple as possible and still meeting your security requirements. In most cases, delegating administrative permissions in Active Directory falls under one of the following scenarios:

### Assigning full control of one OU

- This is a fairly common scenario when a company has multiple offices with local administrators in each office who need to manage all objects in the local office. This option may also be used for companies that have merged multiple resource domains into OUs in a single Active Directory domain. The former resource domain administrators can be given full control of all objects in their specific OU. Using this option means that you can almost completely decentralize the administration of your organization while still maintaining a single domain.

### Assigning full control of specific objects in an OU

- This is a variation on the first scenario. In some cases, a company may have multiple offices, but local administrators should have permission to manage only specific objects in the office OU. For example, you may want to allow a local administrator to manage all user and group objects, but not computer objects. In a situation where resource domains have become OUs, you may want OU administrators to manage all computer accounts and domain-local groups in their OU, but not to manage any user objects.

### Assigning full control of specific objects in the entire domain

- Some companies have highly centralized user and group administration, where only one group has permission to add and delete user and group accounts. In this scenario, this group can be given full control of user and group objects regardless of where the objects are located within the domain. This is also a fairly common scenario for a company with a centralized workstation and server administration group. The workstation team may be given full control of all computer objects in the domain.

### Assigning rights to modify only some properties for objects

- In some cases, you may want to give an administrative group permission to manage a subset of properties on an object. For example, you may want to give an administrative group permission to reset passwords on all user accounts, but not to have any other administrative permissions. Or, the Human Resources department may be given permission to modify the personal and public information on all user accounts in the domain, but not permission to create or delete user accounts.

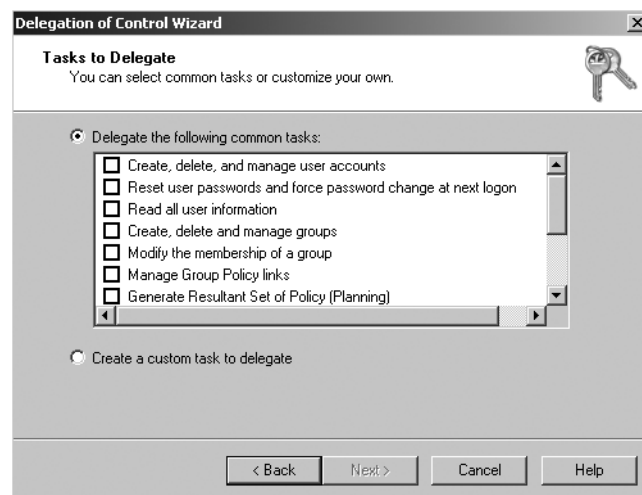
It is possible to use all of these options, and any combination of these options, with Windows Server 2008 AD DS. As mentioned previously, one way to configure delegated permissions is by directly accessing the ACL for an object and configuring the permissions.

The problem with this option is that it can get quite complex because of the number of options available and the real possibility of making a mistake.

To make this task easier, AD DS includes the Delegation of Control Wizard.

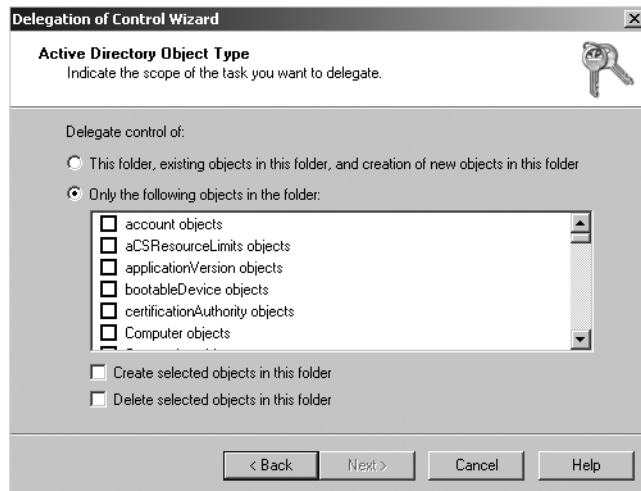
### To use the Delegation of Control Wizard, follow these steps:

1. Open the **Active Directory Users and Computers** administrative console and identify the parent object where you want to delegate control. In most cases, you will be delegating control at an OU level, but you can also delegate control at the domain or container level (for example, the Computers or Users container). Right-click the parent object and click **Delegate Control**. Click **Next**.
2. On the **Users or Groups** page, add the users or groups to which you want to delegate control. Click **Add** to search Active Directory for the appropriate users or groups.
3. Next, select the tasks that you want to delegate. The interface (shown in Figure 9-13) enables you to select from a list of common tasks or to create a custom task to delegate.



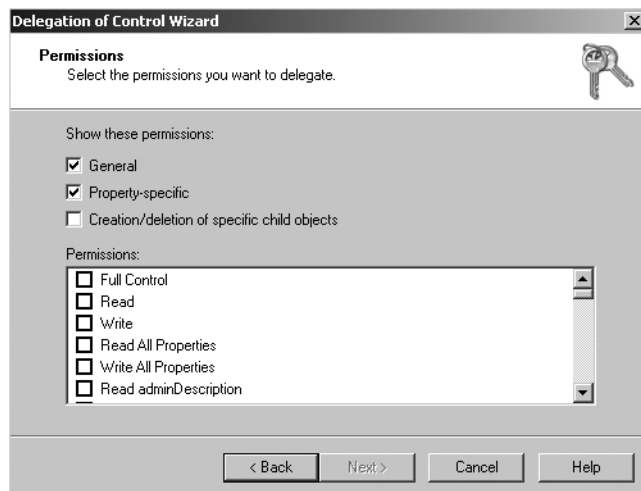
**Figure 9-13** Using the Delegation of Control Wizard to select a common task or create a custom task to delegate.

4. If you choose to create a custom task, you can choose the type or types of objects to which you want to delegate administrative permissions. (Figure 9-14 shows the interface.)



**Figure 9-14** Selecting the type of object or objects to which permissions will be delegated.

5. After you have selected the type of object to which to delegate permissions, you can choose what levels of permissions you want to apply to the object. You can choose full control over the object, or you can delegate permissions to specific properties. (The interface is shown in Figure 9-15.)



**Figure 9-15** Selecting the specific permissions to delegate.

The Delegation of Control Wizard makes it much easier to delegate control in a consistent manner than when configuring permissions through the ACL. However, the effect of either method is the same; that is, the ACL on the objects is modified to provide the appropriate level of access.

## Auditing the Use of Administrative Permissions

Delegating administrative tasks in AD DS results in the need to be able to monitor and audit the use of administrative permissions throughout the directory structure. Auditing serves at least two primary purposes. First of all, it provides evidence for changes that have been made to the directory. If a change has been made to the directory, you may

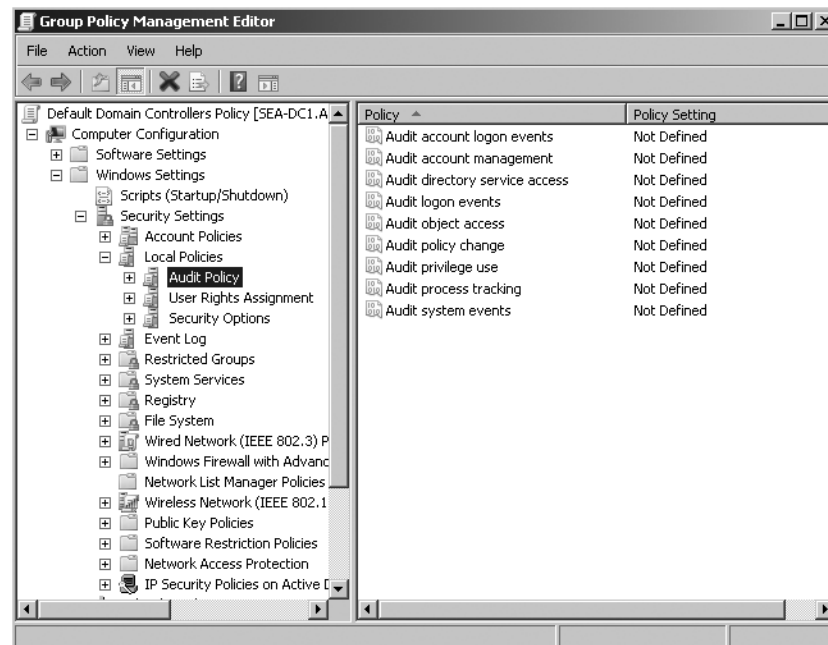
need to track down who has made the change. This is especially important if an incorrect or malicious change has been made to the domain information. A second purpose for auditing is to provide an additional check on the administrative permissions being exercised throughout the domain. By examining audit logs occasionally, you can determine whether someone who should not have administrative rights is in fact exercising such rights.

When AD DS events are audited, entries are written to the Security log on the domain controller. You can then use the Event Viewer to view events that Windows Server 2008 logs in the Security log. You can also save events to an event file that can be used to archive and track trends over time.

There are two steps involved in enabling auditing of changes made to Active Directory objects; configuring the audit policy for domain controllers and configuring the SACL on specific Active Directory objects which are to be audited. These two steps are discussed in the following sections.

## Configuring the Audit Policy for the Domain Controllers

Your first step for enabling auditing is to configure the audit policy for the domain controllers. This can be configured on the Default Domain Controllers Policy found within the Group Policy Management console. When you open the Group Policy Management console, browse to the Group Policy Objects container. In the details pane, you can then right-click Default Domain Controllers Policy and then click Edit to open the Group Policy Management Editor. From the Group Policy Management Editor, you can browse to Computer Management\Windows Settings\Security Settings\Local Policies and then click Audit Policy. Figure 9-16 shows the default auditing configuration in Windows Server 2008 AD DS.



**Figure 9-16** Configuring auditing on the Default Domain Controllers OU.



To audit changes to Active Directory objects, you need to enable and configure the *Audit directory service access* policy. When this policy is enabled and configured, all modifications made to Active Directory objects are reported in the Security Log. You can audit both successful changes to Active Directory objects and failed attempts at modifying Active Directory objects.

In Windows 2000 Server and Windows Server 2003, the *Audit directory service access* policy was the primary option used to audit directory service events. Windows Server 2008 divides this policy into four subcategories:

- Directory Service Access
- Directory Service Changes
- Directory Service Replication
- Detailed Directory Service Replication

Dividing the Audit directory service access policy into four sub-categories provides more granular control on what is or is not audited in relation to directory service events. Enabling the *Audit directory service access* policy enables all the directory service policy subcategories. To modify the subcategories, you cannot use the Group Policy Object Editor. You can only view and modify the subcategories with the command-line tool Auditpol.exe. For example, if you want to view all of the possible categories and subcategories, type the following line at the command prompt, and then press Enter:

```
auditpol /list /subcategory:*
```

[Note] For a list of commands that can be used with Auditpol.exe, open a command prompt and type the following: **Auditpol.exe /?**

---

## Auditing Changes to Objects Using the Directory Service Changes Subcategory

The Directory Services Changes subcategory provides the ability to audit changes to objects in AD DS. This subcategory audits the following types of changes:

- When a modify operation is successfully performed on an attribute, AD DS logs both the previous and current values of the attribute.
- When a new object is created, all values of the attributes that are populated during the creation are logged. Note that any default values to attributes that are assigned by AD DS are not logged.
- When an object is moved within the domain, both the previous and new location is logged.
- If you undelete an object, the location where the object is moved to is logged as well as any additions or modifications to attributes while performing the undelete operation.

To enable the Directory Service Changes audit subcategory, you can type the following line at the command prompt, and then press ENTER:

```
auditpol /set /subcategory:"directory service changes" /success:enable
```

When you enable the Directory Services Changes audit subcategory, AD DS logs various types of events in the Security event log as shown in Table 9-3:

**Table 9-3 Directory Services Changes Events**

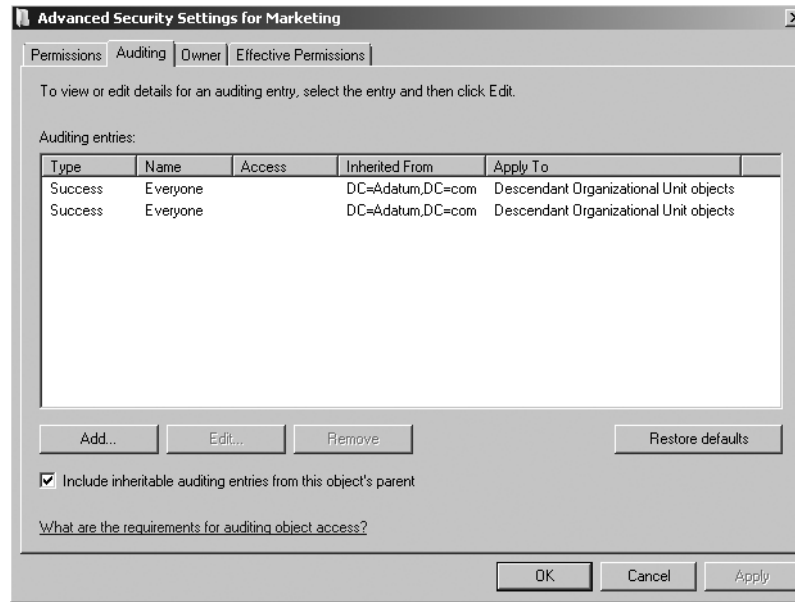
Event ID	Type	Description
5136	Modify	Logged when a modification is made to an attribute in AD DS.
5137	Create	Logged when a new object is created in AD DS.
5138	Undelete	Logged when an object is undeleted in AD DS.
5139	Move	Logged when an object is moved within the domain.

---

## Configuring Auditing on Active Directory Objects

The second step to configuring Active Directory object auditing is to enable auditing directly on the SACL of each Active Directory object to be audited. To enable Active Directory object auditing, access the object's Properties sheet through the appropriate Active Directory administrative tool. Then click the Security page, click Advanced, and click the Auditing page. Figure 9-17 shows the interface for the Active Directory Users and Computers administrative console and the default audit setting for an OU in Active Directory.

To add additional auditing entries, click Add and select which users or groups and what actions you want to audit. In most cases, you should select the Everyone group so that modifications made by anyone can be audited. Then you can select which activities you want to audit. You can audit all modifications made to any object in the container, to specific types of objects, or to specific properties. You can enable the auditing of all successful modifications, of all failed attempts to make modifications, or both. If you audit all successful modifications, you will have an audit trail for all changes made to the directory. If you enable failed attempts, you will be able to monitor any illicit attempts to modify directory information. Once auditing is enabled, all of the audit events are recorded in the Security log accessible through the Event Viewer.



**Figure 9-17** Configuring auditing on Active Directory objects.

Enabling auditing is easy. Managing auditing is much more difficult. If you enable the auditing of all directory modifications at the domain controller OU level, the Security log will grow very rapidly. Almost all of the events will be legitimate changes and thus of no interest to you except as an audit trail. However, interspersed among the legitimate changes may be a small number of changes that you need to be aware of. The problem is finding the few interesting audit events among the large number of routine events. In some companies, one administrator may be given the task of reviewing the event logs every day. A better way to deal with this is to create some automated way of centralizing and analyzing the event logs. Another way is to use a tool such as Microsoft System Center Operations Manager (a separate product available for purchase) to filter the events and raise alerts only on the interesting events.

[Note] If you want to find out more about Microsoft System Center Operations Manager, you can go to the following web site:  
<http://www.microsoft.com/systemcenter/opsmgr/default.msp>. Operations Manager provides a great deal of functionality that goes far beyond just monitoring security logs.

## Tools for Delegated Administration

AD DS provides powerful options for delegating administrative tasks and assigning only the precise permissions that users need to have to perform specific tasks. To complement this delegation, Windows Server 2008 also makes it easy to develop administrative tools that fit the user's task. For example, if you delegate the right to reset passwords for a single OU, you can also provide a very simple administrative tool that can only be used to reset passwords in the specified OU. Windows Server 2008 provides the ability to create a customized view of the Microsoft Management Console (MMC) administrative snap-in in order to allow delegated administrators effective tools to complete their tasks.

## Customizing the Microsoft Management Console

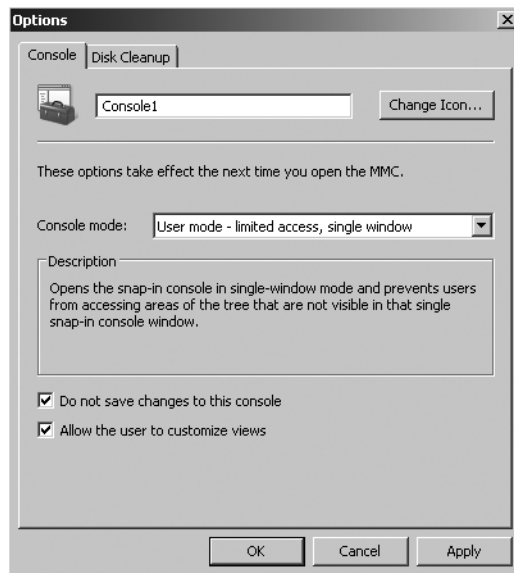
One option for developing an administrative tool is to create a customized MMC using one of the default snap-ins and then modify what the user can see in the MMC.

**[Caution]** Simply creating the customized MMC does not grant or limit the user's rights to perform administrative tasks. Before creating the customized administrative interface, you must first delegate the correct level of permissions. For example, if you give a user the right to create user accounts at a domain level, and then you create an MMC that only allows the user to view one OU, the user can still create user accounts in any OU in the domain. If the user loads the regular Active Directory Users And Computers administrative tool or sits down at another desk with a different MMC, the user will be able to create the account anywhere.

To create the customized MMC, open the Run dialog box and type *mmc*. This opens an empty MMC. From the File menu, add the appropriate Active Directory administrative tool snap-in. If you create a custom MMC using the Active Directory Users and Computers snap-in, you would then expand the domain and locate the container object where you have delegated permissions. In the left pane, right-click on the container object and select **New Window From Here**.

This opens a new window with just the container object and all child objects visible. You can then switch back to the window that displays the entire domain and close the window. Then save the administrative tool and provide it to the users, who will administer only the part of the domain that is visible in the MMC. The MMC can be provided to the user in a number of ways. For example, you may install the MMC on his or her desktop, or you may create a shortcut to the administrative tool on a network share.

To make sure that the administrators do not modify the custom MMC after you have given it to them, you can modify the MMC options by selecting **Options** from the File menu. You can configure the MMC to be saved in **User Mode** and modify the permissions on the MMC so that the end user cannot save any changes to the MMC. Figure 9-18 shows the interface. For full details on how to create customized MMCs, see *Windows Help and Support*.



**Figure 9-18** Configuring a custom MMC to prevent changes to the MMC.

## Planning for the Delegation of Administration

As shown in this chapter, Windows Server 2008 AD DS provides the tools you need to delegate administrative permissions in your domain. However, with all of the positive things you can do in delegating permissions, you also take the risk of assigning incorrect permissions. Incorrect permissions may result in allowing users to do things in Active Directory that they should not be able to do. Incorrect permissions can also mean assigning too few permissions, so that users cannot do the work they need to do. Creating a delegation structure that will provide users with the precise permissions they need requires a significant amount of planning. The following are several suggestions to help with your administrative delegation planning:

- Carefully document the administrative requirements for all potential administrators. In most companies, you will find that there are various users and groups that need some administrative permissions in the domain. Many of these users could be members of the Domain Admins group. As you document the administrative tasks that users need to perform, you will usually find that they really need a much lower level of access. Often the only way to document the level of administrative permissions each group needs is to document all of the administrative work they do every day. By documenting the activities they have to perform, you can design the precise permissions they need to have.
- Before making any changes to the production environment, test all security modifications in a test environment. Making a wrong security configuration can have serious implications for your network. Use the test lab to ensure that the modifications meet the permission requirements, but do not give any additional permissions that are not needed.
- Use the Effective Permissions page in the Advanced Security Settings window to monitor and test the permissions users have. The Effective Permissions page is an effective tool that can be used to determine the precise permissions a user or group

has in AD DS. Use the tool in the test environment to ensure that your configuration is accurate, and use it again in the production environment to make sure that your implementation followed the plan.

- Document all the permissions that you assign. Of all the tasks assigned to network administrators, documenting changes made to the network seems to be the most disliked because it can be very tedious and not seen as important. As a result, documentation is often incomplete or out-of-date. The only way to effectively manage the security configuration on your network is to document the initial configuration and then to make a commitment to keep the documentation updated whenever one of the original settings is modified.

## Summary

The option to delegate administrative permissions in Windows Server 2008 AD DS provides a great deal of flexibility in how your domain can be administered. The delegation of administrative rights is based on the Active Directory security model, where every object and every attribute on every object has an ACL that controls what permissions security principals have to the object. According to the security model, all permissions are, by default, inherited from container objects to objects within the container. These two basic features of the security model mean that you can assign almost any level of permission to any Active Directory object. This flexibility can also mean a great deal of complexity if the security for Active Directory is not kept as simple as possible. This chapter provided an overview of security permissions, Active Directory object access, delegation of administration, and how to audit changes made in Active Directory.

## Additional Resources

The following resources contain additional information and tools related to this chapter.

### Related Information

- Chapter 5, "Designing the Active Directory Structure" provides details on planning the structure of Active Directory such as site, domain, organizational unit, and Forest designs.
- Chapter 6, "Installing Active Directory Domain Services" provides details on delegating administration for Read-Only domain controllers.
- Chapter 8, "Active Directory Security" provides additional details on Active Directory security basics and authentication.
- "Best Practices for Delegating Active Directory Administration":  
<http://www.microsoft.com/technet/prodtechnol/windowsserver2003/technologies/directory/activedirectory/actdid1.mspx>
- "Delegating Authority in Active Directory":  
<http://www.microsoft.com/technet/technetmag/issues/2007/02/ActiveDirectory/default.aspx>
- <more to be added>

## On the Companion Media

- The whitepaper "Best Practices for Delegating Active Directory Administration"
- The whitepaper "Best Practices for Delegating Active Directory Administration: Appendices"
- <more to be added>