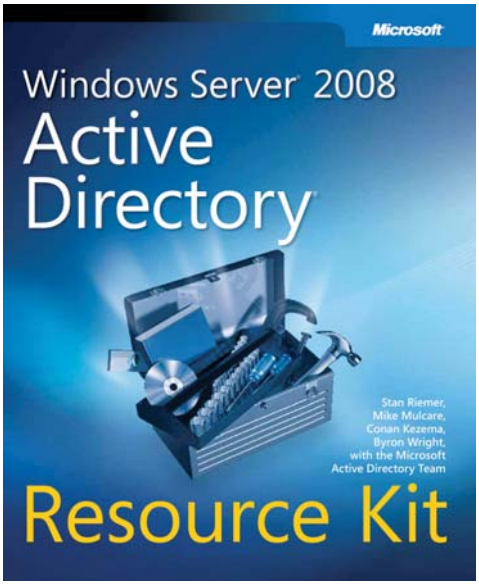


Windows Server® 2008 Active Directory® Resource Kit

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

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

Table of Contents

PART II DESIGNING AND IMPLEMENTING WINDOWS SERVER 2008 ACTIVE DIRECTORY

5 Designing the Active Directory Structure

Defining Directory Service Requirements

Defining Business and Technical Requirements
Documenting the Current Environment

Designing the Forest Structure

Forests and Active Directory Design
Single or Multiple Forests
Designing Forests for Active Directory Security
Defining Forest Ownership
Forest Change Control Policies

Designing the Integration of Multiple Forests

Options for Integrating Forests
Designing Inter-Forest Trusts
Designing Directory Integration Between Forests

Designing the Domain Structure

Domains and Active Directory Design
Determining the Number of Domains
Designing the Forest Root Domain
Designing Domain Hierarchies
Domain Trees and Trusts
Changing the Domain Hierarchy
Defining Domain Ownership
Designing Domains for Active Directory Security

Designing the DNS Infrastructure

Examining the Existing DNS Infrastructure
Namespace Design

Designing the Organizational Unit Structure

Organizational Units and Active Directory Design
Designing an OU Structure
Creating an OU Design

Designing the Site Topology

Sites and Active Directory Design
Networking Infrastructure and Site Design
Creating a Site Design
Designing Domain Controller Locations

Designing Active Directory Domain Controller Configurations

Considerations for Designing Domain Controllers
Hardware Recommendations for Domain Controllers
Combining Services on Domain Controllers
Designing Security for Domain Controllers

6 Installing Active Directory Domain Services

Prerequisites for Installing Active Directory

Hard Disk
Network Connectivity
DNS
Administrative Permissions

Active Directory Installation Options

Server Manager
Active Directory Installation Wizard (Dcpromo.exe)
Unattended Installation

Using the Active Directory Installation Wizard

Performing an Unattended Installation

Scenarios for Using Unattended Installations
Unattended Installation Options

Advanced Installation Options

Deploying Read-Only Domain Controllers

Installing RODCs
Installing an RODC on Server Core
Preparing a Staged Installation
Configuring Password Replication Policies
Configuring Delegated Administrators

Configuring Time Synchronization

Verifying and Troubleshooting Domain Controller Installation

Removing Active Directory

Removing Additional Domain Controllers
Removing the Last Domain Controller
Unattended Removal of Active Directory
Forced Removal of Active Directory

7 Upgrading to Windows Server 2008 Active Directory

Migration Paths

The Domain Upgrade Migration Path
The Domain Restructure Migration Path
The Upgrade-Then-Restructure Migration Path

Determining Your Migration Path

Migration Path Decision Criteria
Choosing the Domain Upgrade Path
Choosing the Domain Restructure Path
Choosing the Upgrade-Then-Restructure Path

Preparing for Migration to Active Directory

Planning the Migration
Testing the Migration Plan
Conducting a Pilot Migration

Upgrading the Domain

Upgrading from Windows 2000 Server
Upgrading from Windows Server 2003

Restructuring the Domain

Creating the Pristine Forest
Migrating Account Domains
Migrating Resource Domains

Upgrading then Restructuring

PART III Administering Windows Server 2008 Active Directory

8 Active Directory Security

Active Directory Security Basics

Security Principals
Access Control Lists
Access Tokens
Authentication
Authorization

Implementing Secure Authentication

Introduction to Kerberos
Kerberos Authentication
Delegation of Authentication
Configuring Kerberos in Windows Server 2008
Integration with Public Key Infrastructure
Integration with Smart Cards
Interoperability with Other Kerberos Systems
NTLM Security

Managing Published Shared Folders

Automating Active Directory Object Management

Command Line Tools for Active Directory Management
Using LDIFDE and CSVDE
Using VBScript to Manage Active Directory Objects
Using Windows PowerShell to Manage Active Directory Objects

11 Introduction to Group Policies

Group Policy Overview

Group Policy Components

Group Policy Processing

How Clients Process GPOs
Initial GPO Processing
Background GPO Refreshes
GPO Processing on Slow Network Links
Loopback Processing

Implementing Group Policies

Creating GPOs
Administering Group Policy Objects
Group Policy Inheritance and Application
Modifying the Default Application of Group Policies
Delegating Administration of GPOs
Implementing Group Policies Between Domains and Forests

Managing Group Policies Using Group Policy Management Console

GPMC Overview
Managing GPO Links
Copying and Importing GPOs
Backing Up and Restoring GPOS
Group Policy Reporting
Group Policy Modeling

Additional Group Policy Management Tools

RSoP Tool
GPResult
GPUpdate

Scripting Group Policy Management

Delegating Control of Group Policy Objects

Options for Delegating Control of GPOs
Delegate control to create GPOs
Delegate control of manage GPO links
Delegate control to generate group policy modeling and results
Delegate control to configure WMI filters

Restoring Sysvol Information
Restoring Operations Masters and Global Catalog Servers

Part 6: Identity and Access Management with Active Directory

16 Active Directory Lightweight Directory Services

AD LDS Overview

AD LDS Features
AD LDS Deployment Scenarios

Implementing AD LDS

Installing AD LDS
Configuring Instances and Application Partitions
Manage AD LDS Objects
Configuring Replication

Configuring the Integration of AD LDS and AD DS

Scripting AD LDS Management

17 Active Directory Certificate Services

AD CS Overview

Public Key Infrastructure Components
Certification Authorities
Certificate Services Deployment Scenarios
AD CS and AD DS Integration

Implementing AD CS

Installing AD CS Root Certificate Authorities
Installing AD CS Subordinate CAs
Configuring Web Enrollment
Configuring Certificate Revocation
Managing Key Archival and Recovery

Managing Certificates in AD CS

Configuring Certificate Templates
Configuring Certificate AutoEnrollment Settings
Automating Certificate Distribution

Designing a AD CS Implementation

Designing a CA Hierarchy
Designing Certificate Templates
Designing Certificate Distribution and Revocation

18 Active Directory Rights Management Services

AD RMS Overview

AD RMS Features
AD RMS Components
AD RMS Processes
AD RMS Deployment Scenarios

Implementing AD RMS

Installing AD RMS Clusters
AD RMS Service Connection Points
AD RMS Clients
AD RMS Databases

Administering AD RMS

AD RMS Certificates
Exclusion Policies
Trust Policies
User Accounts
Rights Policy Templates
Revocation Lists

Designing an AD RMS Implementation

Designing an Intranet Implementation
Designing an Extranet Implementation
Designing an Inter-Organization Implementation

19 Active Directory Federated Services

AD FS Overview

AD FS Features
AD FS Concepts
AD FS Processes
AD FS Deployment Scenarios

Implementing AD FS

Installing AD FS Role Services
Configuring Account Partner ADFS Components
Configuring Resource Partner ADFS Components

Designing an AD FS Implementation

Federated Web SSO Deployment
Federated Web SSO with Forest Trust
Web SSO
Integrating AD FS and AD RMS

Chapter 4

Active Directory Domain Services Replication

In almost all cases, when you deploy an Active Directory Domain Services domain in Microsoft Windows Server 2008 you should deploy more than one domain controller. Deploying multiple domain controllers in each domain is the easiest and most effective way to provide high availability for the domain controller services. These domain controllers might all be located in one data center at the company head office where they are connected by very fast network connections. Or they might be spread across many locations around the world, with a variety of wide area network (WAN) connections linking the company locations.

Regardless of how many domain controllers a company has or where those domain controllers are located, they must replicate information with each other. If they cannot replicate the information, the directories on the domain controllers will become inconsistent. For example, if a user is created on one domain controller and that information is not replicated to all the other domain controllers, the user will be able to log on to only the domain controller where the account was created.

This chapter describes the process of replication in AD DS. The focus of this chapter is on how replication works, that is, on how the replication topology is created and how domain controllers replicate with each other. By default, when you install AD DS domain controllers, they automatically begin replicating with each other. This default replication topology may not be the most efficient for your organization, so this chapter describes ways that you can modify the replication configuration to meet your company requirements. In addition, this chapter provides guidance on how to troubleshoot AD DS replication.

AD DS Replication Model

As described in Chapter 2, "Active Directory Components," AD DS is made up of multiple logical partitions. Replication between the domain controllers with replicas of each partition is handled in exactly the same way for all partitions. When an attribute is changed in the configuration directory partition, it is replicated using the same model and processes as when an attribute is changed in any other partition. The only thing that changes is the list of domain controllers that will receive a copy of the replicated change. Also, replication between domain controllers in the same site is handled differently than it is between domain controllers in different sites, but the essential model does not change. This section describes the replication model used by AD DS.

AD DS uses a multimaster replication model. That means that changes to the AD DS data store can be made on any domain controller except specifically configured read-only domain controllers (RODC). That is, every domain controller except the RODCs has a writable copy of the directory and there no single domain controller where changes have to be made. Once a change has been made, it is replicated to all the other domain

controllers. This multimaster replication model addresses many important reliability and scalability issues. Because all of the domain controllers provide the same services, no domain controller represents a single point of failure.

> [Note] As discussed in Chapter 2, AD DS has specific operations master roles that can be held by only one domain controller. These roles represent a single point of failure, but the roles can also be easily moved or seized to another domain controller.

The replication model used by AD DS can be described as being loosely consistent, but with convergence. The replication is *loosely consistent* because not all domain controllers with a replica of a partition will always have identical information. For example, if a new user is created on one of the domain controllers, the other domain controllers will not receive that information until the next replication cycle. The replication always moves towards *convergence*, however. If the system is maintained in a steady state, with no new changes made to the directory for a period of time, all domain controllers will reach a state of convergence where they all have identical information.

The replication model also uses a *store and forward* replication process. This means that a domain controller can receive a change to the directory and then forward the change to other domain controllers. This is advantageous in a scenario in which multiple domain controllers in a number of company locations are separated by slow WAN links. A change to the directory can be replicated from one domain controller in one site to a single domain controller in another site. The domain controller that receives the update can then forward the changes to other domain controllers in the second site.

AD DS also uses a state-based replication model. This means that each domain controller tracks the state of replication updates. As a domain controller receives new updates (either by changes being made on the domain controller, or through replicated changes from another domain controller), the domain controller applies the updates to its replica of the AD DS data store. When another domain controller attempts to replicate information that a domain controller already has, the receiving domain controller can determine by the state of its data store that it does not need to get the duplicate information. The current state of the data store includes metadata that is used to resolve conflicts and to avoid sending the full replica on each replication cycle.

Replication Process

Features such as multi-master replication and store and forward replication mean that a domain controller could receive AD DS updates from multiple domain controllers and that AD DS replication traffic could take more than one path between domain controllers. For example, if a change is made to AD DS on DC1, the change could be replicated directly to DC2 and DC3. Because of the store and forward replication model, DC2, after receiving the update from DC1, may try to replicate the same change to DC3. AD DS replication is designed to ensure that the replication process is efficient while still providing redundancy.

Update Types

There are two types of changes that can be made to the AD DS information on a particular domain controller. The first type of update is an *originating update*. An originating update is performed when an object is added, modified, or deleted on a domain controller. The second type of update is a *replicated update*. A replicated update is performed when a change that was made on another domain controller is replicated to the local domain controller. By definition, there can be only one originating update performed for any particular change, and this occurs on the domain controller where the change is made. This originating update is then replicated to all the domain controllers that have a replica of the affected AD DS partition.

Originating updates occur in AD DS under any of the following circumstances:

- A new object is added to AD DS. Adding a new object to AD DS creates an object with a unique objectGUID attribute. As well, all values assigned to attributes that are configured for the object are assigned a version number of 1.
- An existing object is deleted from AD DS. When an object is deleted from AD DS, it is marked as deleted, but not immediately removed from the AD DS data store. Only after the deletion has been replicated to all other domain controllers is the object actually deleted. For more details, see the section "Replicating Object Deletions" below.
- The attributes for an existing object are modified. This modification can include adding a new value to an attribute, deleting a value for an attribute, or modifying an existing value. When you change an object, the modify request compares the new value for each attribute with the existing value. If the value for an attribute has not changed, the attribute is not updated. If the value has changed, the version number for each updated attribute is incremented by one.
- An object in AD DS is moved to a new parent container. If the parent container is renamed, each object in the container is also moved to the renamed container. When an object is moved to another container in AD DS, the only attribute that changes for the object is the name attribute, which is changed to reflect the new location in the LDAP hierarchy.

All originating updates to AD DS are *atomic operations*, which means that when an update is committed to AD DS, either the entire transaction is committed and permanent, or no part of the update will be committed. For more information on the process of committing changes to the AD DS data store, see Chapter 14, "Monitoring and Maintaining Active Directory".

The Replication Process in Windows Server 2008

Windows Server 2003 introduced several important changes to the replication process that are also available in Windows Server 2008. One of these changes is the partial replication of multivalued attributes. In Windows 2000, the smallest unit of replication is an attribute. This means that in some cases, changing one value in a multivalued attribute can create a significant amount of replication traffic. The most common example of this is what happens with universal group membership. Because the entire

membership list for the universal group is one attribute, adding a single user to the universal group results in significant replication, especially when the group already had several thousand members. In Windows Server 2003 Active Directory and Windows Server 2008 AD DS, multivalued attributes like group membership can be updated by replicating only the attribute's updated value.

AD DS uses linked attributes to enable replication of individual values of a multivalued attribute. Linked attributes always include a forward link and backward link to create a link between two AD DS objects. The forward link is the linked attribute on the source object (for example, the member attribute on the group object), while the backward link is the linked attribute on the target object (for example, the memberOf attribute on the user object). A backward link value includes the distinguished names of all the objects that have the object's distinguished name set in their corresponding forward link.

The relationships between linked attributes are stored in a separate table in the directory database as link pairs. The matching pair of Link IDs tie the attributes together. For example, the member attribute has a link ID of 2 and the memberOf attribute has a link ID of 3. Because the member and the memberOf attributes are linked in the database and indexed for searching, the directory can be examined for all records in which the link pair is member/memberOf and the memberOf attribute identifies the group.

Another important change in Windows Server 2003 Active Directory is the support for groups of more than 5,000 members. In Windows 2000, groups cannot contain more than 5,000 members because of the attribute-level updates and replication. The practical limit for committing a change to the directory database in one transaction is 5,000. This also defines the maximum number of updates that can be replicated in one update during replication. This means that the maximum group size in Windows 2000 is 5,000 members. In Windows Server 2008 AD DS, support for modifications of only one value on a multivalued object removes these restrictions.

Replicating Changes

After an originating update has been committed to AD DS, the change must be replicated to other domain controllers that host a replica of that partition. Within a site, the domain controller where the originating update occurred waits 15 seconds before replicating the changes to its direct replication partners. The 15-second wait occurs so that if multiple updates are committed to the database, they can all be replicated at the same time. This increases the efficiency of the replication. Between sites, the originating update will be replicated to replication partners based on the schedule configured on the site link.

When replicating changes to the directory information, the domain controllers require a mechanism for managing the flow of replication. To optimize AD DS replication, only those changes that need to be replicated between two domain controllers should be sent. To accomplish this, the domain controllers should be able to determine what, if any, changes have not yet been replicated, and then replicate only those changes that are required. AD DS uses a combination of update sequence numbers (USNs), high-watermark values, up-to-dateness vectors, and change stamps to manage directory replication.

Update Sequence Numbers

When an object is updated in the database, an *update sequence number* is assigned to the update. The USN is specific to the domain controller where the update occurred. For example, if a telephone number update for one user was assigned USN 5555, the next change to the domain controller, regardless of which object was modified, would be USN 5556. One USN is assigned for each committed change. If multiple attributes are changed with one update (for example, a user's address, telephone number, and office location are all modified at once), only one USN is assigned during the update.

There are three ways that the USN is used when an update is committed. First, the local USN value is stored with the attribute that was updated. The local USN value identifies the USN of the changed attribute. The second way the USN is used is for the object's *uSNChanged* attribute. This attribute is stored with each object and identifies the highest USN for any attribute for the object. For example, suppose a user's telephone number was changed and the USN applied to that change was 5556. Both the local USN and the *uSNChanged* attribute will be set to 5556. If the next update applied to the directory on that server were an address change for the same user, the local USN on the address attribute and the *uSNChanged* attribute for the user object would both be changed to 5557. However, the local USN for the telephone number attribute would remain at 5556, because that was the USN for the last update that changed that particular attribute.

The local USN and the *uSNChanged* attribute are applied for both originating and replicated updates. The last way the USN is used is as the *originating USN* for the attribute. This value is set only for originating updates and is replicated to all other domain controllers as part of the attribute replication. When the telephone number for a user is changed on a server, the USN for the change is assigned to the originating USN value. When the modified telephone number is replicated to another domain controller, the originating USN is sent along with the update and this value is not modified on the destination domain controller. The local USN and the *uSNChanged* attribute will be modified on the destination domain controller (and will be specific to that domain controller), but the originating USN is not changed until the attribute itself is updated again. The originating USN is used for propagation dampening, which is described later in this chapter.

Viewing USN Information

The USNs for any object can be viewed through different administrative tools included with Windows Server 2008. The easiest way to view the current and original USN values for an object is to use the Active Directory Users And Computers administrative tool. To view this information, turn on Advanced Features under the View menu and then access the Object tab in the object's Properties sheet. Remember that the USN number is domain controller-specific, so that if you view the USN for an object on two different domain controllers, the USN will be different.

One way to view the local USN, originating domain controller, originating USN, and time stamp for any attribute is by using the Repadmin command-line tool. Type *repadmin /showobjmeta domaincontrollername objectdistinguishedname* at a command prompt. Figure 4-1 shows the partial output from this command.

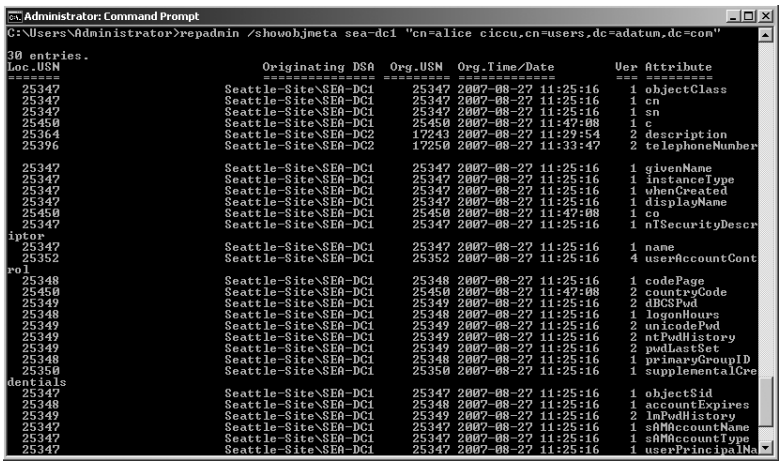


Figure 4-1 Viewing replication meta-data using Repadmin.

In this output, you can see that the user was created on SEA-DC1, but then the description and telephoneNumber attributes were modified on SEA-DC2. The originating USNs for all of the attributes except these two are from SEA-DC1, while the originating USNs for the description and telephoneNumber attributes are from SEA-DC2. However, the local USN numbers are all from SEA-DC1, which is the domain controller where this information was captured. As well, the version number for these two attributes is 2, indicating that the attribute has been modified from the original version.

You can also access the same replication information through Ldp.exe. To do this, connect and bind to a domain controller using LDP, locate the object and then right-click the object, select Advanced, and then select Replication Metadata. The replication metadata is the same information as is shown in the Repadmin tool except that the originating DSA information is shown using the domain controller GUID rather than the display name.

High-Watermark Values

The high-watermark values are used to manage what information is replicated between domain controllers. Each domain controller maintains its own set of high-watermark values for each of its direct replication partners. The high-watermark is just the latest *uSNChanged* value that the domain controller has received from a specific replication partner. When a domain controller sends an update to a replication partner, the *uSNChanged* value is sent along with the update. The destination domain controller retains this *uSNChanged* as the high-watermark value for the replication partner.

The high-watermark values are used during the process of replication. When one domain controller requests updates from another domain controller, the destination domain controller sends its high-watermark value for use by the sending domain controller. In effect, the high-watermark is telling the sending domain controller which updates the destination domain controller has already received. The sending domain controller uses the destination domain controller's high-watermark to filter all of the potential directory updates and sends only the changes with a higher *uSNChanged* value.

[Note] A separate high-watermark value is maintained for each directory partition on the domain controller and for each direct replication partner.

Up-To-Dateness Vectors and Propagation Dampening

The up-to-dateness vectors are also used to control what information is replicated between domain controllers. The up-to-dateness vectors are used to keep track of all of the originating updates that a domain controller has received from any domain controller. For example, suppose the telephone number for a user is changed on DC1 and the attribute is given the originating USN of 5556. When this attribute is replicated to DC2, the originating USN is replicated with the updated attribute. Also, the server GUID for DC1 is replicated with the attribute. When DC2 receives this update, it will modify its up-to-dateness vector to show that the latest originating update it received from DC1 is now 5556.

When a destination domain controller requests updates from a sending domain controller, it includes its up-to-dateness vectors with the request. The sending computer then uses this information to filter the list of all possible updates it could send to the destination domain controller. This option is important when there are more than two domain controllers for a directory partition. For example, if DC3 is added to the scenario described in the preceding paragraph, the telephone number change made on DC1 will be replicated to both DC2 and DC3. Now both DC3 and DC2 will have the updated telephone number, and they will modify their up-to-dateness vector to show that the latest update both of them received from DC1 had an originating USN of 5556. About 15 seconds after receiving this update, DC2 will notify DC3 that it has updated information. When DC3 requests the directory updates from DC2, it will include its up-to-dateness vector with the request. In this case, DC2 determines that DC3's up-to-dateness vector for DC1 already has the most recent originating USN. If this telephone number update were the only change made to the directory during this time period, no information would be replicated between the DC2 and the DC3 domain controllers.

This process of limiting the updates sent during replication by using the up-to-dateness vector is called *propagation dampening*. This is an important feature because AD DS is designed to create redundant replication connections between domain controllers. One of the problems with creating the redundant links is that the same updates might be sent to a domain controller from multiple replication partners. This could create a significant amount of unnecessary replication traffic, as well as potentially leading to a situation where the same update is sent repeatedly to all domain controllers (resulting in a replication loop). Propagation dampening using the up-to-dateness vector eliminates this possibility.

The high-watermark and up-to-dateness vector are used together to limit replication traffic. The high-watermark identifies the latest change that a domain controller received from another specific domain controller, so the sending domain controller does not need to resend changes. The up-to-dateness vector identifies the most recent changes that have been received from all other domain controllers that contain a replica of the partition, so that the sending domain controller does not have to send any directory

updates that the receiving domain controller has received from another replication partner.

Change Stamps and Conflict Resolution

The last property that is used to manage the replication between domain controllers is a *change stamp*. Whenever an attribute is updated, this modification is marked with the change stamp. The change stamp is then sent with the update when it is replicated to other domain controllers. The change stamp is used to determine which change will be accepted in the case of a replication conflict. The change stamp consists of three components:

Version number

- This is used to track the number of changes that have been made to an attribute on an object. When an object is created, the version number on all attributes is set to 0 if the attribute is left blank. When a blank attribute is assigned a value, the version number is incremented to 1. Whenever the attribute is updated the version number increments by one each time.

Last write time

- This is used to track when the last write occurred to the attribute. The time value is recorded on the server where the attribute is updated and is replicated with the object to other domain controllers.

Originating server

- This is the GUID for the server where the last originating update to the attribute was applied.

These three components form the change stamp for every modification to an attribute. When the attribute is replicated to another domain controller, this change stamp information is replicated with the attribute. If the same attribute is changed on two different domain controllers at the same time, this change stamp is used to determine which attribute is accepted as the final change. If a conflict arises, the decision as to which is the final change is made in the following order:

1. Version number. The change with the highest version number is always accepted. This means that if the change on one domain controller is version 3, and the change on the other domain controller is version 4, the version 4 change will always be accepted.
2. Last write time. The next value used to determine which value is accepted is the last write time. If the version numbers are identical, the change with the most recent time stamp will be accepted.
3. Server GUID. If the version numbers are identical and the timestamps are identical, the server database GUID is used to determine which change is accepted. The change coming from the server with the higher GUID will be accepted. These GUIDs are assigned when the domain controllers are added to the domain and the assignment of the GUID is arbitrary.

Replication Conflicts in the Real World

Some network administrators seem to get very concerned about the possibility of replication conflicts and the potential for lost or overwritten data. In most companies,

the chances of a replication conflict happening are slim. First, replication conflicts are dealt with at a per-attribute level. (If a user's telephone number is changed on one domain controller at the same time that the user's address is changed on another domain controller, no conflict is created.) Second, most companies have a centralized department where all changes to user accounts are made, so the chances of two people making different changes to the same attribute at the same time are remote. If the administration of user accounts is delegated to a department level, each department would make changes only to the user accounts for their department. So for most companies with a structured way of working with AD DS objects, replication conflicts should occur rarely.

The AD DS replication process is able to resolve conflicts that are created when the same attribute on an object is modified on two domain controllers at the same time. However, there are at least two other types of conflicts that can arise:

- Adding an object or modifying an object on one domain controller at the same time that the container object for the object is deleted on another domain controller. Take the example in which on one domain controller a new user is added to the Accounting organizational unit (OU). At the same time, on another domain controller, another administrator deletes the Accounting OU. In this case, the container will be deleted on all domain controllers through replication, and the object that was added to the deleted container will be moved to the LostAndFound container in AD DS.
- Adding objects with the same relative distinguished name into the same container. An example of this conflict is when an administrator on one domain controller creates a user object with a relative distinguished name of Bill in the Accounting OU and at the same time, on another domain controller, a user with the same relative distinguished name is moved into the same OU or created in the same OU. In this case, the conflict resolution model will use the GUID assigned to the directory updated to determine which object is kept and which object is renamed. The object with the higher GUID is retained, and the object with the lower GUID is renamed to Bill*CNF:*userGUID,* where the number sign (*) is a reserved character. If the second user object is required, it can be renamed.

Replicating Object Deletions

The replication of object deletions is handled differently in AD DS than other directory updates. When an object like a user account is deleted, the object is not immediately deleted. Rather, a tombstone object is created. The *tombstone object* is the original object with the *isDeleted* attribute on the object set to *true*, and most of the attributes for the object are removed from it. Only a few attributes that are required to identify the object such as the GUID, SID, USN, and distinguished name are retained. Deleted objects are stored in the Deleted Objects hidden container. Every directory partition has a Deleted Objects container.

[Note] To view the Deleted Objects container in a directory partition, use a tool like LDP.exe. After connecting and binding to the directory partition, access the Controls option on the Options menu. In the Controls dialog box, add the Return Deleted Objects control. After

adding the control, you will be able to view the CN=Deleted Items container when you view the directory tree.

This tombstone is then replicated to other domain controllers in the domain. As each domain controller receives the update, the modifications that were made on the originating domain controller are applied to each domain controller. The tombstone objects remain in the domain database for a specified period of time, called the *tombstone lifetime*. At the end of the tombstone lifetime, set to 180 days by default, each domain controller removes the tombstone from its copy of the database. This process of removing the tombstones from the database is called *garbage collection*. By default, the garbage collection interval for the forest is set at every 12 hours. This means that every 12 hours, the garbage collection process runs and deletes any tombstones that have passed the tombstone lifetime value.

[Note] The default tombstone lifetime in versions of Active Directory before Windows Server 2003 Service Pack 1 was 60 days. If you upgrade an existing domain to Windows Server 2008, the 60 day tombstone lifetime is retained. The tombstone lifetime and the garbage collection interval can be modified using ADSI Edit or Ldp.exe. These properties are configured on the CN=Directory Service,CN=Windows NT,CN=Services,CN=Configuration, DC=*ForestRootDomain* object. The *garbageCollPeriod* and the *tombstoneLifetime* attributes define these settings. In most cases, these values do not need to be modified.

Linked attributes can result in special cases when deleting objects. When an object is deleted, the following changes are made to the linked attributes:

- All of the forward links on the deleted object are removed. For example, if a group object is deleted, all of the member links on the group object are removed. This means that the group is removed from the memberOf back-link attribute on each user that was a member of the group.
- All the back-links on the deleted object are removed. For example, if a user is deleted, the user's distinguished name value is removed from the member attributes of each group object that is named in the memberOf attribute of the deleted user.

After the linked attribute has been modified on one domain controller, the updates are replicated to other domain controllers just like any other updates.

[Important] Because of how linked attributes are deleted, you have to treat the authoritative restore of these objects differently than if you are restoring objects without linked attributes. For details, see Chapter 15, "Active Directory Disaster Recovery."

Replicating the Sysvol Directory

Changes to the AD DS data store are made using the process described above. However, the SYSVOL folder on each domain controller also contains information that is critical to the correct functioning of AD DS. The SYSVOL shared folder contains the following files and folders that must be available and synchronized between domain controllers in a domain:

- Group policy settings. SYSVOL contains a folder with the name of the domain that the domain controller is a member of. In the domain folder is a folder called Policies which contains group policy templates and scripts for Windows 2000 or later clients.
- The NETLOGON shared folder, which includes system policies (Config. pol or Ntconfig. pol files) and user-based logon and logoff scripts for pre-Windows 2000 network clients, such as clients running Windows 98 or Windows NT 4.0. The NETLOGON shared folder is the Scripts folder in the domain folder.

The contents of SYSVOL folder are replicated to every domain controller in a domain. If the domain is at Windows Server 2003 or lower functional level, the File Replication Service (FRS), is responsible for replicating the contents of the SYSVOL folder between domain controllers. When you upgrade the domain functional level to Windows Server 2008, Distributed File System Replication (DFSR) is used to replicate the contents of the SYSVOL folder. In both cases, the connection object topology and schedule that the Knowledge Consistency Checker (KCC) creates for Active Directory replication is used to manage replication between domain controllers.

> [Note] DFS Replication is a state-based, multimaster replication engine introduced in Windows Server 2003 R2 that supports replication scheduling and bandwidth throttling. DFS Replication uses a new compression algorithm that is known as Remote Differential Compression (RDC). Using RDC, DFS Replication replicates only the differences (or changes) between the two servers, resulting in lower bandwidth use during replication. For more information on DFSR, see the article "Overview of the Distributed File System Solution in Microsoft Windows Server 2003 R2" at *http://technet2.microsoft.com/windowsserver/en/library/d3afe6ee-3083-4950-a093-8ab748651b761033.mspx?mfr=true*.

Intrasite and Intersite Replication

The description of how AD DS replication works applies to both intrasite and intersite replication. In both cases, the domain controllers use the same processes to optimize the replication process. However, one of the main reasons to create additional sites in AD DS is to manage replication traffic. Because all of the domain controllers within a site are assumed to be connected with fast network connections, replication between these domain controllers is optimized for maximum speed and reduced latency. However, if the replication traffic has to cross a slow network link, conserving network bandwidth is a much more significant issue. Creating multiple sites allows for this conservation of network bandwidth by enabling features such as data compression and scheduling AD DS replication.

Intrasite Replication

The primary goal for replication within a site is to reduce replication latency, that is, to make sure that all domain controllers in a site are updated as quickly as possible. This means that intrasite replication traffic has the following characteristics:

- The replication process is initiated by a notification from the sending domain controller. When a change is made to the database, the sending computer notifies a destination domain controller that changes are available. The changes are then pulled

from the sending domain controller by the destination domain controller using a remote procedure call (RPC) connection. After this replication is complete, the domain controller notifies another destination domain controller, which then pulls the changes. This process continues until all the replication partners have been updated.

- Replication occurs almost immediately after a change has been made to the AD DS information. By default, a domain controller will wait for 15 seconds after a change has been made and then begin replicating the changes to other domain controllers in the same site. The domain controller will complete replication with one partner, wait 3 seconds, and then initiate replication with another partner. The reason the domain controller waits 15 seconds after a change is to increase the efficiency of the replication in case additional changes are made to the partition information.
- The replication traffic is not compressed. Because all the computers within a site are connected with fast network connections, the data is sent without compression. Compressing the replication data adds an additional load on the domain controller server. By not compressing the replication traffic, server performance is preserved at the expense of network utilization.
- Replication traffic is sent to multiple replication partners during each replication cycle. Whenever a change is made to the directory, the domain controller will replicate the information to all direct replication partners, which might be all or some of the other domain controllers in the site.

Modifying Intrasite Replication

In most cases, you will not need to modify how replication works within in a site. However, there are some settings that you can modify in specific situations. These settings include:

- If your AD DS forest is running in Windows Server 2003 or Windows Server 2008 functional level, you can modify the time that the domain controller will wait before notifying the first replication partner and before notifying subsequent replication partners. To do this, open the Configuration partition in ADSIEdit and browse to the CN=Partitions folder. In the folder, right-click the partition where you want to modify the replication settings. The value for the delay in notifying the first replication partner is stored in the msDS-Replication-Notify-First-DSA-Delay attribute. The default value is not displayed but is set at 15 seconds. The value for subsequent notification delay is stored in the msDS-Replication-Notify-Subsequent-DSA-Delay attribute. The default value is 3 seconds. If your organization contains Windows 2000 Server domain controllers, you must modify the registry on the Windows 2000 Server domain controllers to modify the default settings of 300 seconds to notify the first replication partner and 30 seconds for subsequent notifications.
- You can also configure strict replication consistency. Strict replication consistency determines how outdated objects are replicated from reconnected domain controllers that have not replicated in longer than a tombstone lifetime. For example, if a domain controller is offline while an object is deleted, and remains offline for the entire tombstone period, the tombstone is never replicated to the server. When the server is reconnected to the network, it will try to replicate the object to other domain controllers. If the destination domain controller has strict replication consistency

enabled, it will not accept the inbound replication of an outdated object is blocked. By default, Windows Server 2008 enforces strict replication consistency. You can modify this by setting the value of the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters\Strict Replication Consistency key to 0.

- You can modify the amount of data that is replicated in each replication packet. By default, the number of objects Windows Server 2008 domain controllers will replicate in a single packet is 1/1,000,000th the size of RAM, with a minimum of 100 objects and a maximum of 1,000 objects. The maximum size of objects that will be replicated is 1/100th the size of RAM, with a minimum of 1 megabyte (MB) and a maximum of 10 MB. You can modify these settings by creating the Replicator intra site packet size (objects) and Replicator intra site packet size (bytes) values in the HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\NTDS\Parametersppath.

Intersite Replication

The primary goal of replication between sites is to reduce the amount of bandwidth used for replication traffic. This means that intersite replication traffic has the following characteristics:

- Replication is initiated according to a schedule rather than when changes are made. To manage replication between sites, you must configure a site link connecting the two sites. One of the configuration options on the site link is a schedule for when replication will occur. Another is the replication interval setting for how often replication will occur during the scheduled time. If the bandwidth between company locations is limited, the replication can be scheduled to happen during non-working hours.
- Replication traffic is compressed down to about 40 percent of the non-compressed size when replication traffic is more than 32 KB in size. To save bandwidth on the network connection, the bridgehead servers in each site compress the traffic at the expense of additional CPU usage.
- Notifications are not used to alert a domain controller in another site that changes to the directory are available. Instead, the schedule determines when to replicate.

> [Note] You can disable compression for intersite replication and enable notifications. For more details, see the section Configuring Intersite Replication later in this chapter.

- Intersite replication connections can use either an Internet Protocol (IP) or a Simple Mail Transfer Protocol (SMTP) transport. SMTP can be used as a transport protocol only for the configuration, schema and application directory partitions, not for the domain partition. The connection protocol you use is determined by the available bandwidth and the reliability of the network that connects company locations.
- Replication traffic is sent through bridgehead servers rather than to multiple replication partners. When changes are made to the directory in one site, the changes

are replicated to a single bridgehead server (per directory partition) in that site, and the changes are then replicated to a bridgehead server in the other site. The changes are replicated from the bridgehead server in the second site to all the domain controllers in that site.

- You can easily modify the flow of replication between sites. Almost every component of the intersite replication can be changed.

> [Important] One of the key elements in designing AD DS is site design. Site design includes planning the number and location of sites plus the configuration of intersite connections to optimize the use of network bandwidth while minimizing the replication latency. Configuration options for the intersite connections are discussed later in this chapter, while site design issues are discussed in Chapter 5, "Designing the Active Directory Structure."

Replication Latency

Because of the way replication works in Windows Server 2008 AD DS, it can take some time for a change made on one domain controller to be replicated to all the other domain controllers in an organization. This time lag is called the *replication latency*. In most cases, the replication latency is easy to calculate, especially within a site. As mentioned earlier, any change made to the data store on one domain controller will be replicated to that domain controller's replication partners in about 15 seconds. The destination domain controller will hold that change for 15 seconds and then pass it on to its replication partners. So the replication latency within a site is about 15 seconds times the number of hops the change has to take before reaching all domain controllers. As explained in the next section, the replication topology within a site never requires more than three hops, so the maximum replication latency within a site will usually be less than one minute.

Determining the replication latency between sites is more difficult. First of all, you must calculate the replication latency within the source site. This replication latency is the amount of time it takes for a change made on a domain controller in the site to be replicated to the source site's bridgehead server. Once the information arrives at the originating site's bridgehead server, the site link schedule and replication interval determine the amount of time it takes for the information to get to the destination site. The default configuration for site links is to replicate every 3 hours. If this configuration is not changed, a maximum of 3 hours will be added to the replication latency. When the information arrives at the bridgehead server in the destination site, the intrasite replication latency for the destination site must be added. In some cases, this replication latency might be unacceptable. To minimize this, you can shorten the replication interval to a minimum of 15 minutes for intersite replication.

Managing replication latency is a matter of balancing the need for a short latency period and bandwidth limitations. If you want the shortest possible latency period, you should put all the domain controllers in the same site, and the replication latency will be about one minute for all domain controllers. However, if your company locations are separated by WAN connections with limited bandwidth, you will require multiple sites so that you can manage network utilization for AD DS replication, but replication latency will be higher.

Urgent Replication

In some cases the replication latency described in the previous section is too long. In particular, this is the case when a security-related attribute has been modified in the directory. For these situations, AD DS uses *urgent replication,* in which a domain controller forwards the changes immediately to its replication partners. Any domain controller receiving an urgent update will also immediately forward the change. In this way, all domain controllers in the site are updated within seconds. The following types of changes trigger an urgent replication.

- Modifying the account lockout policy for the domain
- Modifying the domain password policies
- Moving the relative identifier (RID) master to a new domain controller
- Changing a Local Security Authority (LSA) secret, such as when the domain controller machine password is modified
- Locking out a user account. This happens when a user attempts to logon too many times using an incorrect password.
- Changing the relative identifier (RID) master role owner

By default, urgent updates apply only to intrasite replication and not to intersite replication. This default handling of urgent updates can be modified by enabling notification for replication between sites.

User password changes are not replicated using the same urgent replication model. Instead, when a user changes his or her password on a domain controller, the password change is immediately replicated directly to the PDC emulator for the domain. This replication crosses site boundaries and does not make use of the bridgehead servers in each site. Instead, the domain controller where the change was made uses an RPC connection to the PDC emulator to update the password. The PDC emulator then updates all the other domain controllers through the normal process of replication. If the user tries to log on to a domain controller that has not yet received the new password, the domain controller will check with the PDC emulator to see if there are any updated password changes for the user before denying the logon.

Replication Topology Generation

One of the keys to understanding AD DS replication is understanding how the replication topology is created. By default, the process of creating the replication topology is handled automatically by AD DS. While the replication topology can be manually configured, in most cases the default configuration by the system is the best option.

In order for the replication topology to be successfully created, the following components must be in place:

- Routable IP infrastructure. In order to configure intersite replication, you need to configure AD DS sites and map the sites to IP subnet address ranges. Domain controllers and client computers use this IP subnet to site mapping when locating domain controllers.

- DNS. AD DS replication topology requires DNS in order for domain controllers locate replication partners. DNS also stores SRV resource records that provide site affinity information to clients searching for domain controllers.
- Net Logon service. Net Logon is required for DNS registrations.
- Remote Procedure Call (RPC) connectivity. AD DS domain controllers must be able to connect to other domain controllers in the same domain by using RPCs. RPCs must be used between domain controllers in the same site and in different sites if the domain controllers are in the same domain. SMTP is an alternative protocol that can be used by domain controllers in different domains and sites.
- Intersite Messaging. Intersite Messaging is required for SMTP intersite replication and for site coverage calculations. If the forest functional level is Windows 2000, Intersite Messaging is also required for intersite topology generation.

Knowledge Consistency Checker

KCC is the process that runs on every domain controller and is responsible for creating the replication topology within a site and between sites. As soon as a domain controller is added to an AD DS forest, KCC begins creating a replication topology that is both efficient and fault tolerant. As additional domain controllers are added to a site, or as additional sites are added, KCC uses the information about servers, sites, site links, and schedules to create the optimal replication topology.

The KCC runs on each domain controller. On each domain controller, the KCC uses the forest information stored in the configuration directory partition to create a replication topology. Because all domain controllers use the same configuration information and use the same algorithm for creating the topology, the topology is created without the KCC components on different domain controllers directly communicating with each other. The KCC communicates with other KCCs only to make an RPC request for replication error information.

KCC also dynamically deals with changes or failures within the replication topology. If one of the domain controllers is offline for a period of time, KCC revises the replication topology to work around the unavailable domain controller. By default, KCC on every domain controller recalculates the replication topology every 15 minutes. You can force KCC to recalculate the replication topology at any time through the Active Directory Sites And Services administrative tool by locating the server where you want to check the replication topology, right-clicking the NTDS Settings container in the server container, selecting All Tasks, and then selecting Check Replication Topology.

Connection Objects

When KCC creates the replication topology, it creates a series of connection objects that are stored in the configuration directory partition of AD DS. The connection objects are direct logical connections between domain controllers that are used to replicate directory information. KCC tries to create a replication topology that is both efficient and fault tolerant. KCC builds as many connection objects as are required to achieve these goals.

Connection objects are always created as one-way pull connections between two domain controllers. This is because the normal process of replication is always a pull operation where the destination domain controller requests the information from a sending domain controller. In most cases, KCC will build two one-way connections between domain controllers so that information can be replicated either way.

In most cases, the connection objects automatically created by KCC are optimized and you do not need to make any changes. However, in some cases you might want to modify the connection objects. For example, you might want to ensure that the operations master domain controllers in your domain are always direct replication partners with the domain controllers that you have designated as your fallback operations masters in the case of an operations master failure. By creating a connection object between the two domain controllers, you can ensure the optimal replication topology for that particular set of domain controllers.

You can modify the default connection objects in two ways: by modifying some settings on connection objects created by KCC and by adding new connection objects.

Modifying a Connection Object Created by KCC

You can modify the schedule and the source domain controller for a connection object within a site, and you can also modify the transport protocol for connection objects between sites. The connection interface is shown in Figure 4-2. By default, domain controllers within a site will check all their replication partners for missed updates every 15 minutes. You can change that schedule to never check or to check every hour or every half hour. When you modify the connection object, it is renamed from *<automatically generated>* to the object's globally unique identifier (GUID). You can rename the object after modifying it.

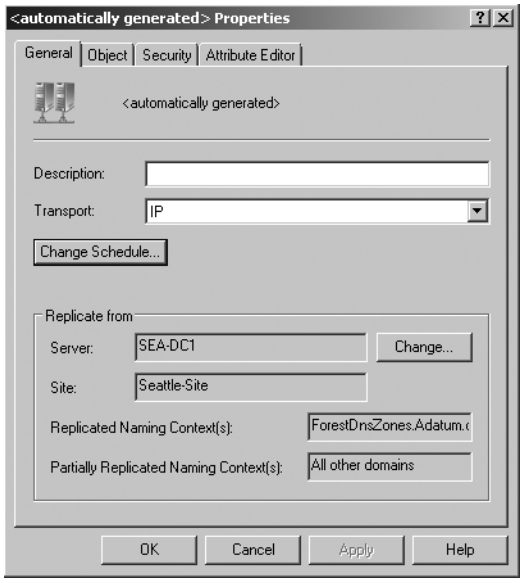


Figure 4-2. Modifying an existing connection object.

Creating a New Connection Object

You can also create an entirely new connection object to force a particular replication topology. When you create a connection object, you are given a choice as to which domain controller to pull changes from. You can also modify any of the other settings on the connection agreement.

KCC will not delete or modify any connections that have been manually modified or created. However, KCC will use the manual connection objects as it would use any other connection, and KCC might reconfigure the connection objects in the site to compensate for the manually created connections.

Intrasite Replication Topology

Within a single site, the KCC will create a replication topology is that includes redundant links. The primary goal for designing AD DS replication is availability and fault tolerance. If a single domain controller is not available for replication, AD DS replication should not fail. The disadvantage of using redundant links is that a domain controller might receive the same update several times because each domain controller will have multiple replication partners. As described earlier, AD DS replication uses propagation dampening to avoid multiple updates of the same information.

As domain controllers with replicas of particular AD DS partitions are added to the organization, KCC automatically begins creating the replication topology. This topology forms a replication ring. Figure 4-3 shows an example of a simple network structure with three domain controllers in the same domain and in a single site.

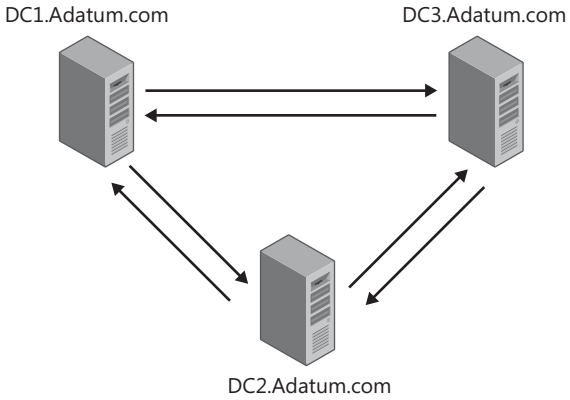


Figure 4-3. A simple replication ring.

As shown in Figure 4-3, KCC creates a replication ring in which every domain controller is configured with two incoming replication connections. If one of the connections is not available, updates can still arrive on the other connection. Also, each domain controller is configured as the source domain controller for two other domain controllers. This creates a redundant ring for each domain controller. As the number of domain controllers with a replica of a particular partition increases, a second principle for creating connections becomes important. KCC will always create a replication topology in which each domain controller in a site is no more than three replication hops away from any other domain controller. As the number of domain controllers with the same directory partition in a site

increases beyond seven, extra connection objects are created to decrease the potential number of hops to three or fewer. For example, the site shown in Figure 4-4 has nine domain controllers. It would have a replication topology that would include at least one additional connection.

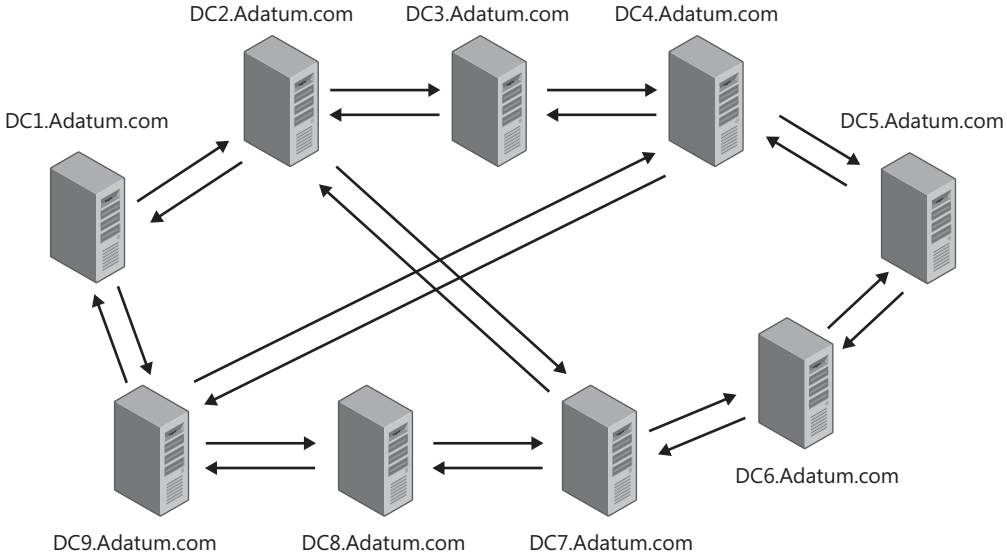
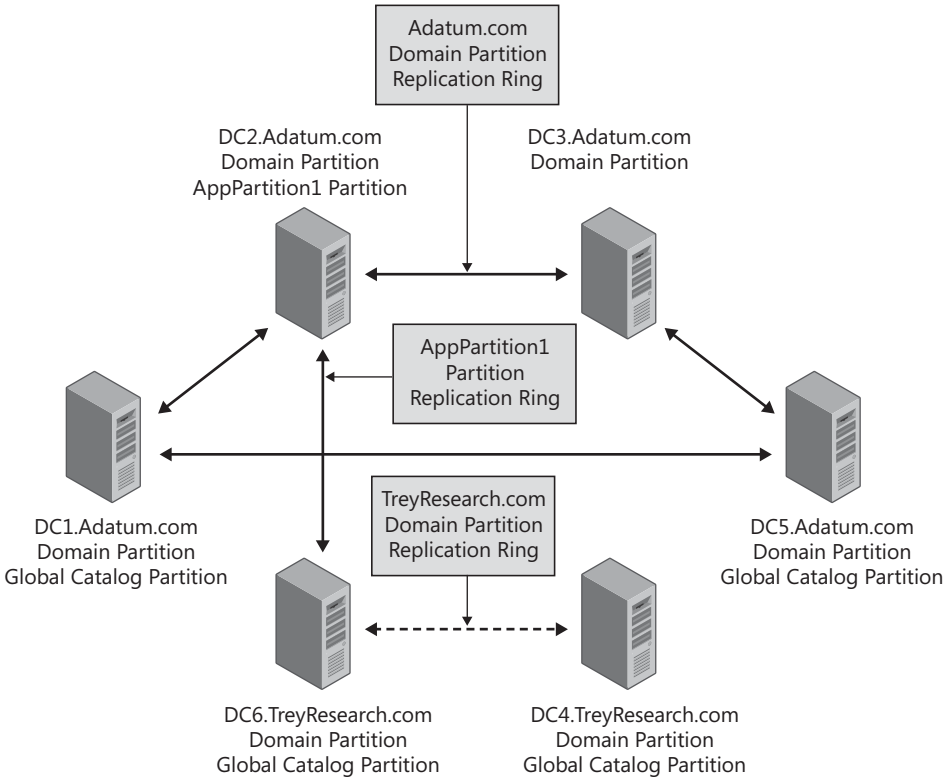


Figure 4-4. A replication ring with more than seven domain controllers.

Replication rings are based on directory partitions. This means that KCC calculates a replication ring for each directory partition. For example, an organization might have multiple domains in a single site and an application directory partition that is replicated to several domain controllers in the site. The configuration could be set up as shown in Figure 4-5.

In the scenario illustrated in Figure 4-5, the replication rings shown in Table 4-1 would be created.



Note: All domain controllers also host a replica of the Configuration and Schema Partitions. The replication ring for the Configuration and Schema Partitions would include all the domain controllers.

Figure 4-5. Replication rings created for each directory partition.

Table 4-1 Replication Rings in a Complex Site

Directory partition	Replication partners
Configuration directory partition, schema directory partition	All the domain controllers would be included in the replication ring for both the configuration directory partition and the schema directory partition
Adatum.com domain directory partition	DC1.Adatum.com, DC2.Adatum.com, DC3.Adatum.com, DC4.Adatum.com.
TreyResearch.com domain directory partition	DC5.TreyResearch.com, DC6.TreyResearch.com.
Global catalog (Global Catalog) partition[1]	DC1.Adatum.com, DC4.Adatum.com, DC5.TreyResearch.com.
AppPartition1 application directory partition	DC2.Adatum.com, DC6.TreyResearch.com.

[Note] The Domain Name System (DNS) application directory partitions (ForestDnsZones and DomainDnsZones) are also included in the replication topology. To keep the Figure 4-5 scenario from getting too complicated, these partitions are not included in that figure nor in the associated table. As discussed in Chapter 3, "Active Directory and Domain Name System," these partitions are treated exactly like other domain directory partitions. Also, the Global Catalog replication topology is not shown in Figure 4-5. The process of creating a Global Catalog replication ring is slightly different than for other partitions and will be described in the next section.

The replication connections and replication status can be viewed by using the Repadmin command line tool with the /showrepl parameter. Figure 4-6 shows the partial output when running this command on a domain controller in a forest with multiple domains and sites.

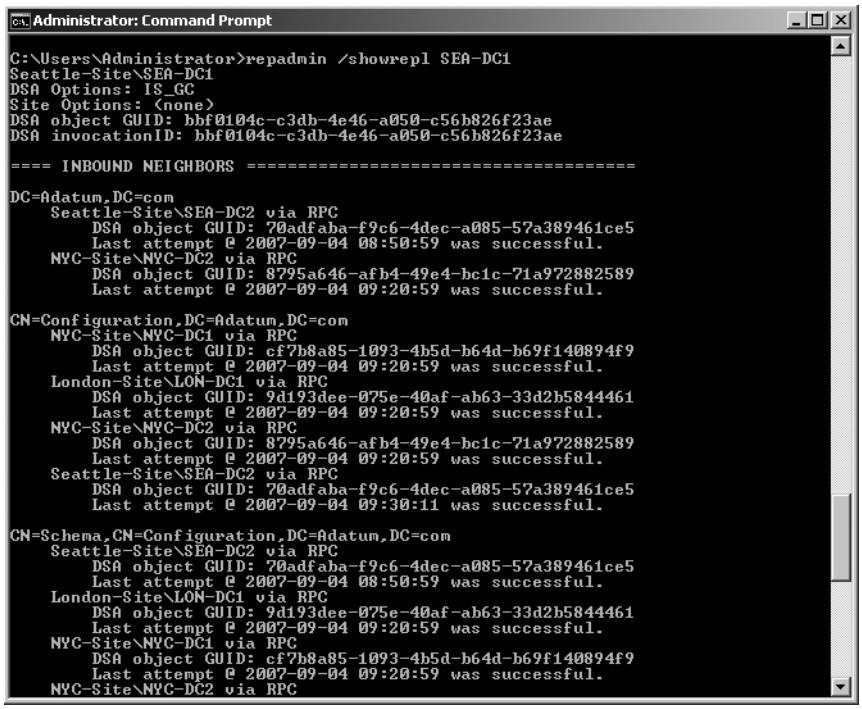


Figure 4-6.

Using Repadmin to view the replication connections.

The replication ring is a logical concept; the actual replication topology as implemented with the connection objects does not duplicate the replication rings exactly. While a separate replication ring is created for each directory partition, KCC will not create additional connection objects for each replication ring. Instead, KCC reuses connection objects to use as few connection objects as possible while still creating a replication topology that provides redundancy for each partition. For example, in the scenario illustrated in Figure 4-5, DC1.Adatum.com has a connection object with DC6.TreyResearch.com. This single connection object could be used to replicate the schema partition, the configuration partition and the AppPartition1 partition as well as the DNS application directory partitions. You can see which directory partitions are replicated by each connection object by viewing the connection object in Active Directory Sites and Services and Repadmin. As is shown in Figure 4-2, you can view the Replicated Naming Context(s) setting on each connection object. You can also use the Repadmin /showconn *servername* to show the partitions that are being replicated by each connection object. Partial output of this command is shown in Figure 4-7. In this figure, you can see that the connection object with SEA-DC2.Adatum.com is being used to replicate the DomainDnsZones, ForestDnsZones, Adatum.com, Configuration and Schema partitions.

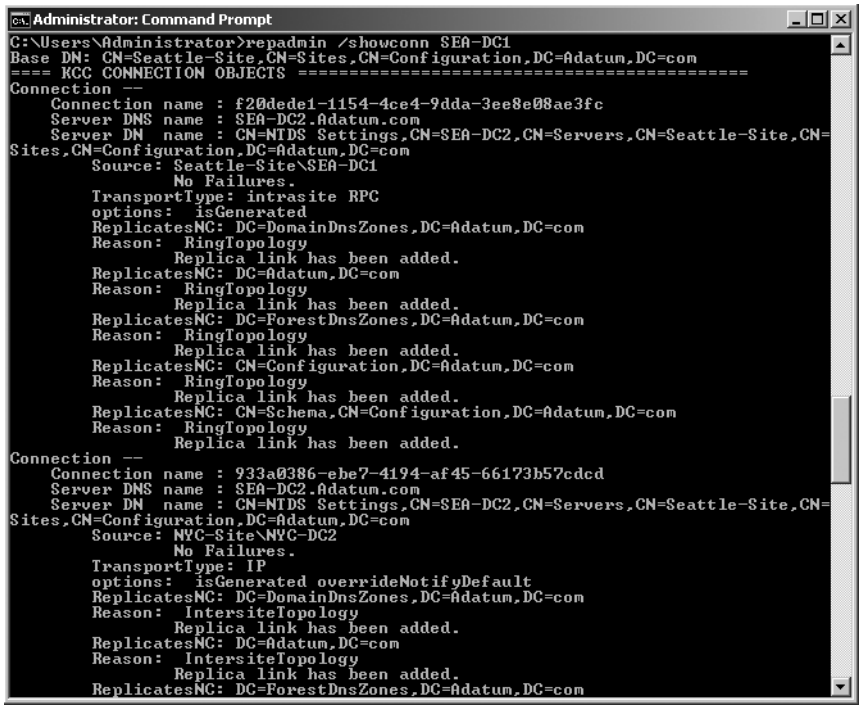


Figure 4-7. A single connection object used to replicate multiple directory partitions.

Global Catalog Replication

The Global Catalog is a different partition than the other partitions in that it is built from all the domain databases in the entire forest. The Global Catalog itself is read-only on all domain controllers, which means that the information in the Global Catalog cannot be directly modified by the administrator. Rather, the Global Catalog is just a list of all the attributes that are moved into the Global Catalog because their *isMemberOfPartialAttributeSet* attribute is set to *true*.

The fact that the Global Catalog is created from the domain databases also affects the replication ring for the Global Catalog. Each Global Catalog server must get the Global Catalog information from the domain controllers in all domains. For a simple example, see Figure 4-8, which shows a company with two domains and one domain controller in each domain, in the same site. Only the DC1.Adatum.com domain is configured as a Global Catalog server. The Global Catalog server is also the only domain controller for the Adatum.com domain, so it will extract the Global Catalog information for Adatum.com from its own domain database. The domain controller in the TreyResearch.com domain has the only copy of that domain directory partition, so DC1.Adatum.com collects the Global Catalog information for the TreyResearch.com domain from DC2.TreyResearch.com. To extract the information from the TreyResearch.com domain, a connection object is created from DC2.TreyResearch.com to DC1.Adatum.com. This connection is then used to replicate the Global Catalog information to DC1.Adatum.com.

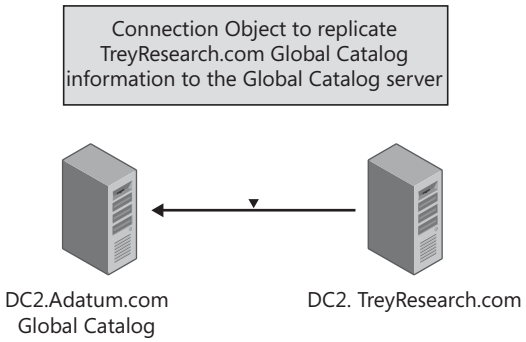


Figure 4-8. An example of simple Global Catalog replication.

Figure 4-9 shows a more complicated example of how the Global Catalog is created and replicated. In this scenario, a connection object is configured from a domain controller in every domain to each Global Catalog server. For example, DC1.Adatum.com will have an inbound connection object from DC2.Adatum.com, DC4.TreyResearch.com, and DC6.Contoso.com. This connection object is used to build the Global Catalog on DC1.Adatum.com. Each of the other Global Catalog servers will have a similar set of connection objects created. Also, a separate replication ring is created for the Global Catalog partition with all of the Global Catalog servers.

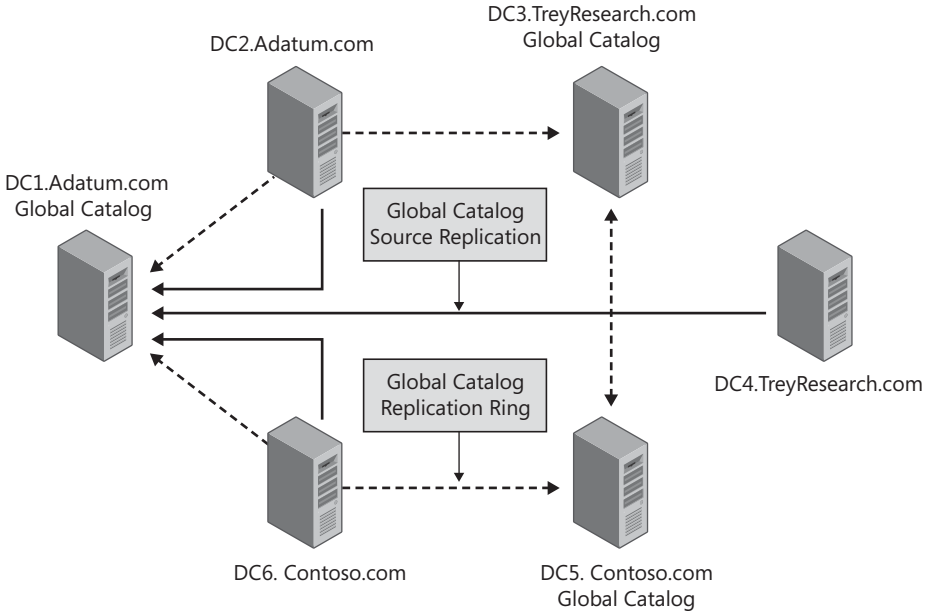


Figure 4-9. An example of a more complicated Global Catalog replication.

Intersite Replication Topology

When additional sites are added to a forest, the replication topology becomes increasingly complex. In a multisite scenario, a replication topology must be created for each site, and a replication topology must be created for replication between sites. To deal with this complexity, the process for creating connection objects changes for the intrasite replication. Within a site, KCC on each domain controller is responsible for creating the connection objects that it needs to ensure that it has the required replication redundancy

for all of its partitions, and it then replicates the information about the connection objects to the other domain controllers. Also, the domain controller receives information about the connection objects that have been created by other domain controllers. The next time KCC runs, connection objects might be added, modified, or deleted based on the information the domain controller has received about other connection objects in the site. Eventually, KCCs on all the domain controllers in a site determine the optimal replication configuration.

A similar approach is used when determining the replication topology between sites, except that one domain controller in each site is responsible for developing the intersite topology. KCC on one domain controller in the site is designated as the Inter-Site Topology Generator (ISTG) for the site. There is only one ISTG per site regardless of how many domains or other directory partitions there are in the site. ISTG is responsible for calculating the ideal replication topology for the entire site. This process consists of the following two actions:

- Identifying the bridgehead servers for each directory partition that is present in the site. Replication between sites is always sent from a bridgehead server in one site to a bridgehead server in another site. This means that information is replicated only once across the network connection between the sites.
- Creating the connection objects between the bridgehead servers to ensure that the information is replicated between the sites. Because the replication is configured between bridgehead servers, there are no redundant connection objects configured as there are within a site. However, the ISTG will create connection objects with bridgehead servers in multiple sites if the site links are configured to enable the connections.

When a new site is added to the forest, ISTG in each site determines which directory partitions are present in the new site. ISTG then calculates the new connection objects that will be needed to replicate the required information from the new site. Also, ISTG designates one domain controller to be the bridgehead server for each directory partition. ISTG creates the required connection agreement in its directory, and this information is replicated to the bridgehead server. The bridgehead server then creates a replication connection with the bridgehead server in the remote site, and replication begins.

To see how the replication topology is created between sites, see Figure 4-10. In this example, the forest contains two sites and two domains with domain controllers for each domain in each site. There is also at least one Global Catalog server in each site. This means that each site contains a directory partition for each of the domains, a Global Catalog partition, as well as the schema directory partition and the configuration directory partition. Two bridgehead servers would be designated in each site, because each of these partitions must be replicated between the sites. One of the bridgehead servers in each site will be a domain controller in the Adatum.com domain. Another bridgehead server in each site must be a domain controller in the TreyResearch.com domain. In the Figure 4-10 example, DC1.Adatum.com and DC6.TreyResearch.com are also Global Catalog servers. This means that they will become bridgehead servers to replicate Global Catalog information between sites. Because the schema directory partition and the configuration directory partition are shared by all domain controllers, one of the existing connection objects can be used to replicate these partitions.

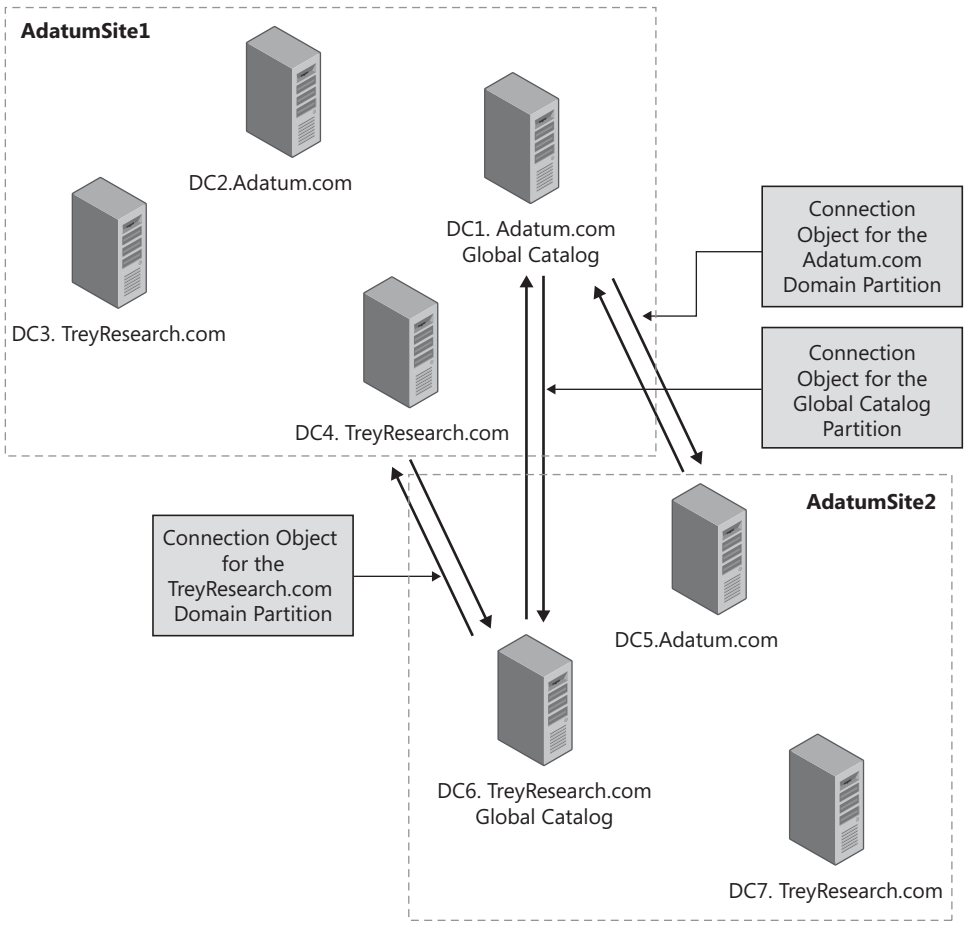


Figure 4-10. Intersite connection objects.

[Note] This discussion of the replication topology is based on the default behavior for AD DS domain controllers. Administrators can modify the default behavior, especially for replication between sites. These modifications and the effect of these changes are discussed later in this chapter.

RODCs and the Replication Topology

RODCs also participate in normal AD DS replication and connection objects must be created between RODCs and other domain controllers. However, because RODCs have read-only copies of the AD DS database, the KCC will only create single one-way connection objects from a domain controller with a writeable copy of the database to the RODC. The RODC can only pull changes from other domain controllers, it can never be configured as a replication source for any connection object.

RODC replication is also limited by which domain controllers can be direct replication partners. RODCs can replicate all AD DS partitions except the domain partition from either Windows Server 2003 or Windows Server 2008 domain controllers. RODCs must replicate the domain partition from a domain controller running Windows Server 2008. This means that each RODC must have a connection object with a Windows Server 2008

domain controller with a writeable copy of the database for the RODC's domain. This also means that when you upgrade a domain from Windows Server 2003, the first Windows Server 2008 domain controller cannot be an RODC.

If the RODC is deployed in a separate site, the Windows Server 2008 should be placed in the nearest site in the topology. Figure 4-11 provides an example of a possible RODC deployment and the connection objects that would be configured.

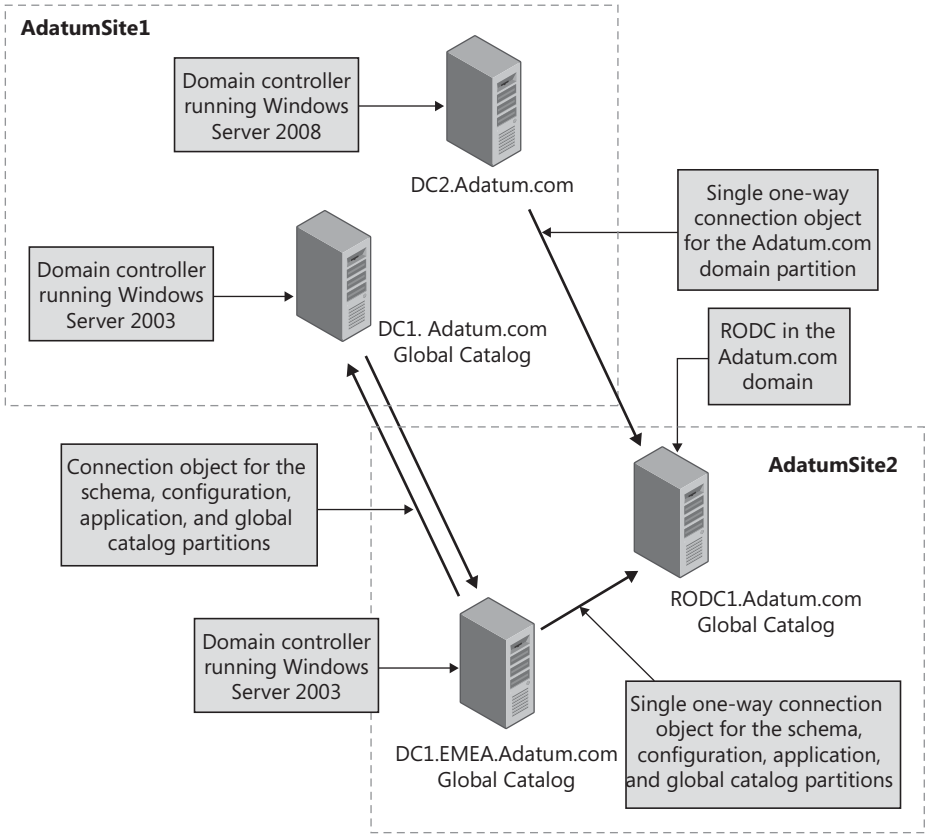


Figure 4-11 Replication connection objects with RODCs.

Configuring Intersite Replication

The most important reason for creating multiple sites in AD DS is to control replication traffic between company locations, especially between locations that are separated by slow WAN connections. As described in Chapter 2, an AD DS site is a network location in which all the domain controllers are connected to each other with fast and reliable network connections. One of the tasks of setting up an AD DS network is determining where to draw the site boundaries and then connecting the sites together.

> [Note] Defining clear criteria for when to create an additional site is difficult because of the large numbers of variables that have to be included in this decision. Chapter 5 goes into detail about when you should consider creating additional sites. That chapter also covers many of the other design issues that you must consider when designing the site topology.

Creating Additional Sites

When AD DS is installed, a single site called the Default-First-Site-Name (the site can be renamed) is created. Because sites are usually based on the company location, you can rename this site to more accurately the location where the domain controllers in the site are located. If additional sites are not created, all subsequent domain controllers will be added to this site as they are installed. However, if your company has multiple locations with limited bandwidth between the locations, you will almost certainly want to create additional sites.

Additional sites are created using the Active Directory Sites and Services administrative tool (see Figure 4-12). To create a new site, right-click the Sites container, and then click New Site.

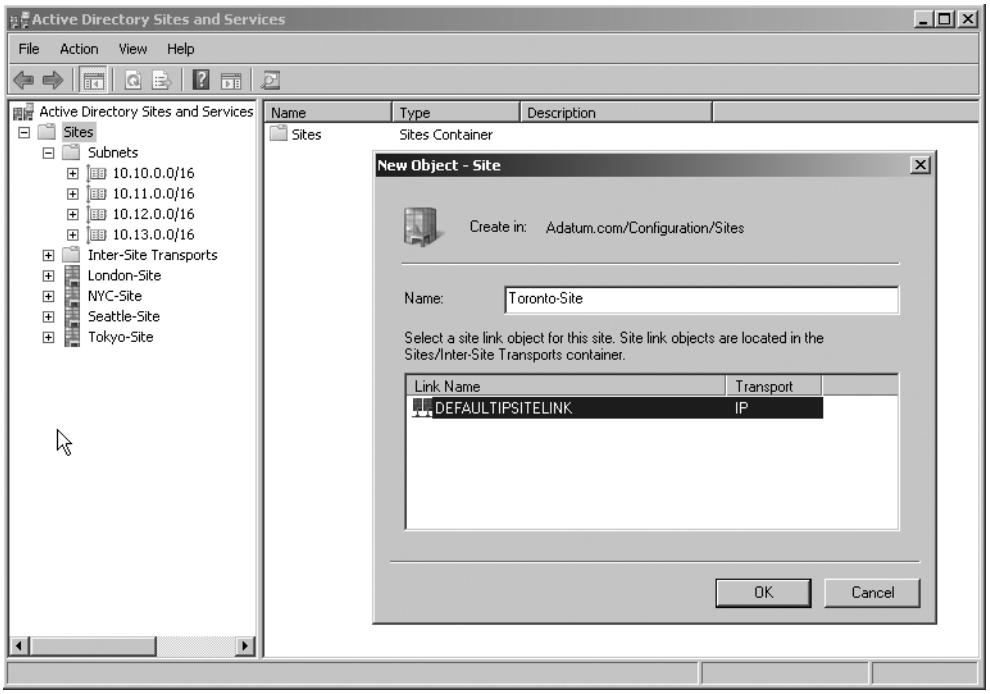


Figure 4-12 Creating a new site in Active Directory Sites and Services.

When you create a new site, you must link the site with an existing site link. This ensures that the site will automatically be included in the replication topology. From the Link Name list, choose which site link will be used to connect this site to other sites.

Each site is associated with one or more IP subnets in AD DS. By default, no subnets are created in AD DS, so you will normally begin by creating all of the subnets associated with the Default-First-Site-Name site. As you create additional sites, also create additional subnets in the Subnets container in Active Directory Sites and Services and associate the subnets with the new site.

[Onthecd] As you modify the site configuration, ensure that you document all of the changes you make. You can use the ADDSSites.xlsx job aid on the CD as a template for documenting the site configuration. You can use the GetADDSSites.ps1 Windows PowerShell script on the CD to display information about all of the sites in your forest. Remember that you need to provide the full path when running a Windows PowerShell script, and you may need to modify the PowerShell script execution policy before you can run a script.

For examples of Visual Basic Scripting Edition (VBScript) scripts that you can use to obtain site information, see the Script Center Script Repository Web site at *http://www.microsoft.com/technet/scriptcenter/scripts/default.mspx?mfr=true*.

Each site should have at least one domain controller and, ideally, a Global Catalog server. To move an existing domain controller into the site, right-click the domain controller object in its current Servers container and select Move. You are then given a choice about which site you want to move the domain controller into. If you install a new domain controller, and you have more than one site in your forest, you are given a choice about which site to install the new domain controller. The default selection in the Active Directory Domain Services Installation wizard is to locate the domain controller in the site where the IP subnet matches the domain controller's IP address.

[Important] When you move a domain controller to a new site, ensure that you modify the domain controller IP configuration to reflect the new site location. You will also need to refresh the host record in DNS by using the IPConfig /refreshDNS command. Verify that you can ping all domain controllers in the domain by fully qualified domain name after completing the domain controller move.

Site Links

The AD DS objects that connect sites together are called *site links*. When AD DS is installed, a single site link—called DEFAULTIPSITELINK—is created. If you do not create any additional site links before you create additional sites, each site is included in this default site link. If all of the WAN connections between your company locations are equal in terms of bandwidth and cost, you can just accept this default configuration. If all the sites are connected by one site link, the replication traffic between all sites will have exactly the same properties. If you make a change on this site link, the replication configuration for all sites will be modified.

[Onthecd] As part of documenting your site configuration, ensure that you also document the site link configuration. You can use the ADDSSites.xlsx job aid on the CD. The ListADDSSites.ps1 script on the CD lists the site links associated with each site.

However, in many cases, you might not want to have the replication between all sites configured the same way. For example, if your company locations are linked by different network connections, you may want to replicate AD DS information across network connections with limited available bandwidth less frequently than you do across network connections with more available bandwidth. If you want to be able to configure different

replication settings between sites, you must create additional site links and assign the appropriate sites to the site links.

> [Note] Creating a site link does not replace the work of ISTG; all it does is make it possible for ISTG to do its work. Once a site link is in place, ISTG will use the site link to create the required connection objects to replicate all the AD DS partitions between each site.

The following are the configuration options on all site links. Figure 4-13 shows the interface for configuring site links.

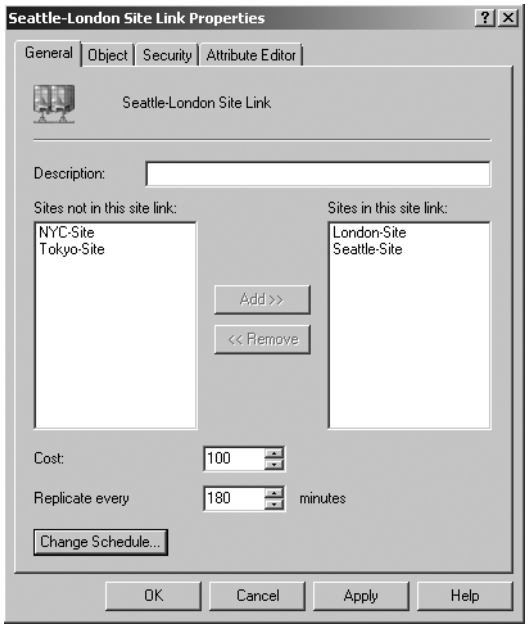


Figure 4-13 Configuring site links.

Cost

- The cost for a site link is an administrator-assigned value that defines the relative cost of the site link. The cost will usually reflect the speed of the network connection and the expenses associated with using the connection. This cost is important if there are redundant site links in the organization, that is, if there is more than one path for replication to travel from one site to another. In all cases, the lowest-cost route will be chosen as the replication path.

Replication schedule

- The replication schedule defines what times during the day the site link is available for replication. The default replication schedule allows for replication to occur 24 hours a day. However, if the bandwidth to a site is very limited, you might want to have replication occur only during non-working hours.

Replication interval

- The replication interval defines the intervals at which the bridgehead servers check with the bridgehead servers in the other sites to see if there are any directory updates. By default, the replication interval for site links is set at 180 minutes. The replication interval is only applied during the replication schedule. If the replication schedule is

configured to allow replication from 10 P.M. to 5 A.M., by default, the bridgehead servers will check for updates every 3 hours during that time.

Replication transports

- The site link can use either RPC over IP or SMTP as the replication transport. See "Replication Transport Protocols" later in this chapter for more details.

These options provide significant flexibility for configuring replication between sites. However, there are also some mistakes to avoid. To understand how these options work together, consider a company network like that shown in Figure 4-14.

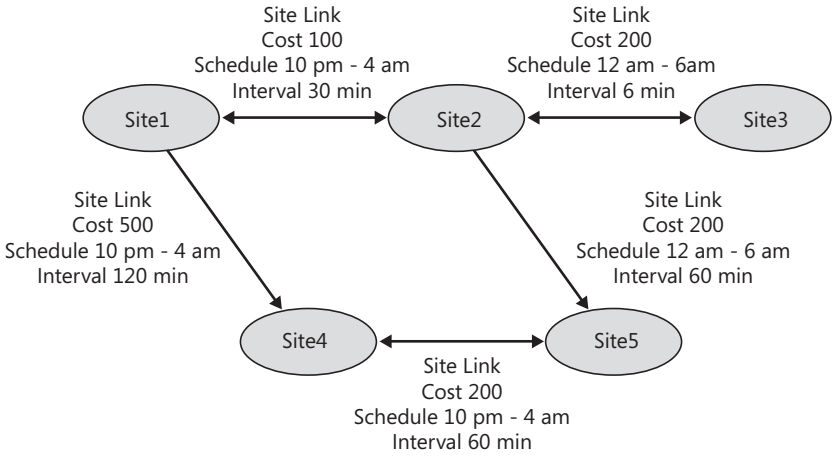


Figure 4-14 A site link configuration.

In Windows Server 2008 AD DS, all site links are considered transitive by default. This means that connection objects can be created between domain controllers that are not in adjacent sites. For example, in Figure 4-14, Site1 has a site link to Site2 and to Site4, and Site2 has a site link to Site3 and Site5. Because of the transitive nature of the site links, this means that domain controllers in Site1 can also replicate directly with domain controllers in Site3 and Site5.

The site link costs define the path that replication traffic will take through the network. When KCC is creating the routing topology, it uses the accumulated costs for all site links to calculate the optimal routing. In the example shown in Figure 4-14, there are two possible routes between Site1 and Site5: The first route is through Site2; the second route is through Site4. The cost to route through Site2 is 300 (100+200) while the cost through Site4 is 700 (500+200). This means that all replication traffic will be replicated through Site2 unless the connection is not available.

When replication traffic crosses multiple site links, the site link schedules and replication intervals for each site link combine to determine the effective replication window and interval. For example, effective replication will occur between Site1 and Site3 only during the hours of 12 midnight to 4 A.M. (the overlapping time in the schedules) and the effective replication will happen every 60 minutes (the replication interval for the Site2–Site3 site link).

[Note] If the schedules for site links do not overlap, it is still possible for replication to occur between multiple sites. For example, if the Site1–Site2 site link is available from 2 A.M. to 6 A.M., and the Site2–Site3 site link is available from 10 P.M.to 1 A.M., changes to the directory will still flow from Site1 to Site3. The changes will be sent from Site1 to Site2, and then from Site2 to Site3. However, the replication latency would be almost a day in this case because changes replicated to Site2 at 2 A.M. would not be replicated to Site3 until 10 P.M.

Additional Site Link Configuration Options

In addition to the site link configuration options available in Active Directory Sites and Services, you can also configure other site link settings by using ADSIEdit, or by modifying the registry on the domain controllers. For example, you configure the following settings:

- Turn off compression for intersite replication. By default, all replication traffic between sites is compressed. However, compressing the traffic places an extra load on the domain controller's processor. If you have sufficient bandwidth between the AD DS sites, you can disable the compression in Windows Server 2008 AD DS.
- Enable notification for intersite replication. By default, replication between sites is based on the schedule and replication frequency configured on the site link. You have the option to enable notification for intersite replication. If notification is enabled, the bridgehead server in a site where a change has occurred notifies the bridgehead server in the destination site, and the changes are pulled across the site link. This can greatly reduce replication latency between sites, but will also increase the network traffic between sites.
  To turn off compression or to turn on notification for intersite replication, you must use a tool such as ADSI Edit to modify the Options attribute on either the site link object or the connection object. To turn off compression, set the value of the Options attribute to *4*; to turn on notification, set the value to *1*.
- You can modify the amount of data that is replicated in each replication packet. By default, the number of objects Windows Server 2008 domain controllers will replicate in a single packet is 1/1,000,000th the size of RAM, with a minimum of 100 objects and a maximum of 1,000 objects. The maximum size of objects that will be replicated is 1/100th the size of RAM, with a minimum of 1 megabyte (MB) and a maximum of 10 MB. You can modify these settings by creating the Replicator inter site packet size (objects) and Replicator inter site packet size (bytes) values in the HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\NTDS\Parametersppath.

Site Link Bridges

In some cases, you might want to turn off the transitive nature of site links and manually configure site link bridges. When you configure site link bridges, you define which site links should be seen as transitive and which site links should not. Turning off the transitive nature of site links can be useful when you do not have a fully routed network, that is, if not all segments of the network are available at all times (for example, if you have a dial-

up or scheduled-demand dial connection to one network location). Site link bridges can also be used to configure replication in situations where a company has several sites connected to a fast backbone with several smaller sites connecting to each larger center using slow network connections. In such cases, site link bridges can be used to manage the flow of replication traffic more efficiently.

> [Note] Chapter 5 provides details about when and how to use site link bridges.

When you create a site link bridge, you must define which site links are part of the bridge. Any site links you add to the site link bridge are considered transitive with each other, but site links that are not included in the site link bridge are not transitive. In the example used earlier, a site link bridge could be created for the site links connecting Site1, Site2, Site4, and Site5. All of these site links would then be considered transitive, which means that a bridgehead server in Site1 could replicate directly with a bridgehead server in Site5. However, because the site link from Site2 to Site3 is not included in the site link bridge, it is not transitive. That means that all replication traffic from Site3 would flow to Site2, and from there it would flow to the other sites.

To turn off the transitive site links, expand the Inter-Site Transport container in Active Directory Sites and Services, right-click IP, click Properties, and then clear the Bridge All Site Links option on the General tab of the IP Properties sheet.

> [Caution] The site link bridging setting affects all site links using the transport protocol where you disable site link bridging. This means that all site link bridging is disabled, and will now have to configure site link bridges for all site links if where you want transitive site connections.

Replication Transport Protocols

Windows Server 2008 AD DS can use one of three different transportation protocols for replication:

- RPC over IP within a site. All replication connections within a site must use an RPC-over-IP connection. This connection is synchronous, which means that the domain controller can replicate with only one replication partner at any one time. The RPC connection uses dynamic port mapping. The first RPC connection is made on the RPC endpoint mapper port (IP port 135). This connection is used to determine which port the destination domain controller is using for replication.

> [Note] If you are replicating the directory information through a firewall, or using routers with port filtering enabled, you can specify the port number that the domain controllers will use for replication. To do this, create the following registry key as a DWORD value and specify any valid port number: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS \Parameters\TCP/IP Port.

- RPC over IP between sites. Replication connections between sites can also use RPC over IP. This RPC connection is the same as the intrasite connection with one important exception: by default all traffic sent between sites is compressed.

> [Note] When you look at the two types of RPC-over-IP connections in the Active Directory Sites And Services administrative tool, you will notice that they are identified differently in the interface. The RPC over IP within a site is called *RPC*, and the RPC over IP between sites is called *IP*.

- SMTP between sites. Replication connections between sites can also use SMTP to replicate information between sites. SMTP can be a good choice as a replication protocol if you do not have a permanent and relatively fast connection between company locations. SMTP uses an asynchronous connection, which means that the domain controller can replicate with multiple servers at the same time.

Configuring SMTP Replication

Configuring SMTP replication is significantly more complicated than configuring RPC over IP replication between sites. With RPC over IP replication, domain controllers use built in components and Kerberos authentication to automatically configure and secure replication.

To configure SMTP replication, you need to complete the following steps:

4. Install the SMTP Server feature on the bridgehead servers in both sites. When you install the SMTP Server feature, required components from the Web Server (IIS) server role are also installed.
5. Install Active Directory Certificate Services and configure the Certification Authority (CA) as an Enterprise CA. The CA will be used to issue certificates to the domain controllers which will be used to sign and encrypt the SMTP messages that are exchanged between domain controllers. When you install an Enterprise CA, it automatically issues domain controller certificates to domain controllers in the same domain as the Enterprise CA. These domain controllers can use the certificates to secure SMTP data. For domain controllers in other domains in the forest, you must manually request a Domain Controller certificate or a Directory Email Replication certificate.
6. Configure SMTP site links with a cost that is less than any RPC over IP site link connecting between the two sites. The two sites must not have any domain controllers in the same domain.
7. Ensure that SMTP e-mail can be sent between the domain controllers. If the domain controllers can communicate directly by using port 25, no further configuration is required. However, in some cases, the domain controllers may need to forward the SMTP messages to a SMTP bridgehead server rather than directly to the destination bridgehead server.

Configuring Bridgehead Servers

As mentioned earlier, replication between sites is accomplished through bridgehead servers. By default, ISTG automatically identifies the bridgehead server as it calculates the

intersite replication topology. To view which domain controllers are operating as bridgehead servers, you can use the Repadmin /bridgeheads command. The command output lists all of the current bridgehead servers in each site, including the directory partitions each bridgehead server is responsible for. The command output also displays whether the last replication with each bridgehead server was successful.

If you run the Repadmin /bridgeheads /v command, the command output displays the last attempted replication for each directory partition on the bridgehead server, as well as the last successful replication time. Figure 4-15 shows the partial output from this command.

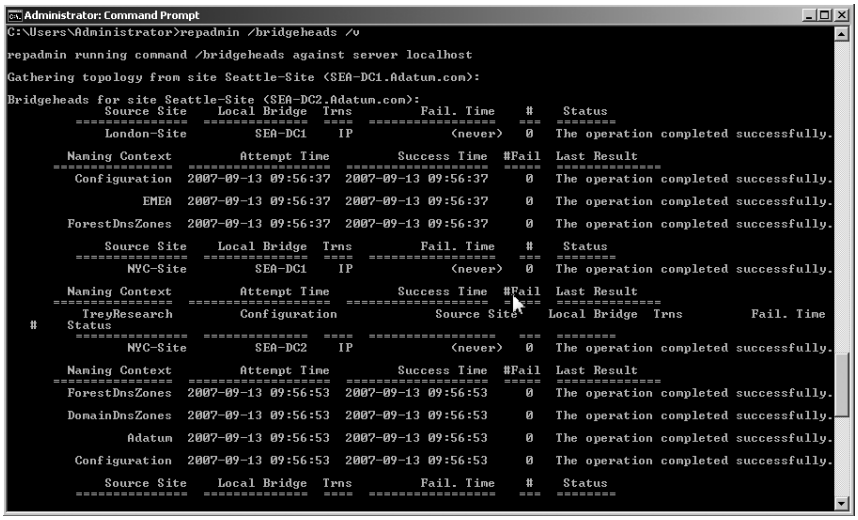


Figure 4-15 Viewing bridgehead server status using Repadmin.

In some cases, you might want to control which domain controllers are going to operate as bridgehead servers. Bridgehead server may require additional server resources if there are many changes to the directory information, replication is set to occur frequently, and the organization has hundreds of sites. To configure which servers will be the bridgehead servers, access the computer objects in the Active Directory Sites and Services administrative tool, right-click the server name, and then select Properties. (Figure 4-16 shows the interface.) You are given the option of configuring the server as a preferred bridgehead server for either IP or SMTP transports.

> [Onthecd] If you configure a bridgehead server and then forget that you configured it, you may spend a lot of time troubleshooting AD DS replication if the bridgehead server fails. Ensure that you document preferred bridgehead servers for both types of replication transports in the ADDSSites.xlsx job aid on the CD.

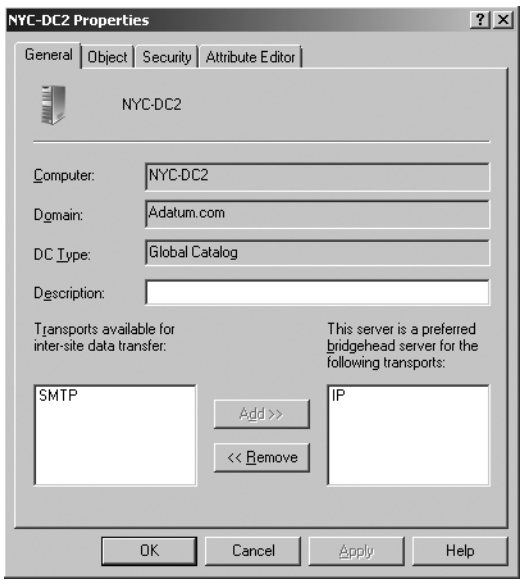


Figure 4-16 Configuring a preferred bridgehead server.

The advantage of configuring preferred bridgehead servers is that you can ensure that the domain controllers you choose will be selected as the bridgehead servers. If you want complete control over which servers are used as bridgehead servers, you must configure a preferred bridgehead server for each partition that needs to be replicated into a site. For example, if a site contains replicas of the Adatum.com domain directory partition, the TreyResearch.com domain directory partition, the Global Catalog partition, and an application directory partition, you will need to configure at least one domain controller with a replica of each of these partitions. If you do not configure bridgehead servers for all of the partitions, you will get a warning message like the one shown in Figure 4-17, and ISTG will log an event in the event log and then choose a preferred bridgehead server for the partition. You can also configure multiple preferred bridgehead servers. If you do, ISTG will choose one of the identified servers as the bridgehead server.

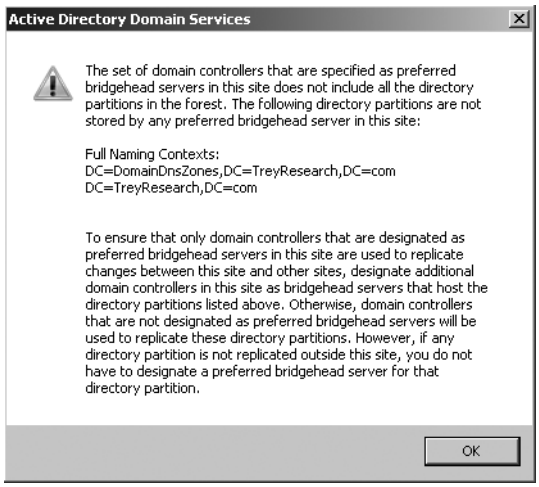


Figure 4-17 Warning message to configure bridgehead servers for each directory partition.

You should configure this option with caution. Configuring preferred bridgehead servers limits ISTG's ability to choose the bridgehead server—it will always select a server that is configured as a preferred bridgehead server. If this server fails and no other servers have been designated as bridgehead servers for that directory partition, ISTG will not select another bridgehead server and replication will cease until the server is again available or until you have reconfigured the preferred bridgehead server options. If the preferred bridgehead server does fail, you can either remove the server as a preferred bridgehead server and allow ISTG to identify a bridgehead server, or you can choose another preferred bridgehead server.

> [Caution] If the preferred bridgehead server does fail, and you choose to reconfigure the preferred bridgehead server, you need to make any configuration changes in both sites. Because the bridgehead servers are not available, no information will be replicated between the sites until the configuration changes are made in both sites. To make changes in a remote site, connect to a domain controller in the site in Active Directory Sites and Services.

Troubleshooting Replication

If AD DS replication fails, domain controllers will not be updated with changes made on other domain controllers. This may lead to inconsistent experiences for users and administrators, depending on which domain controller they are connecting to. If password or configuration changes for users are not replicated, users may not be able to log on to the network. If group policy settings or the Sysvol directory are not replicated, users may experience different group policy settings. Because of the importance of AD DS replication, you should be prepared to troubleshoot AD DS replication issues.

Process for Troubleshooting AD DS Replication Failures

The first step in troubleshooting AD DS replication failures is to identify the reason for the failure. In many cases, it can be difficult to immediately identify why AD DS replication fails, so often troubleshooting is a matter of eliminating possible reasons for failure. As general guidance, complete the following steps:

1. Verify network connectivity. As is the case with most troubleshooting scenarios, start by verifying that the domain controllers can communicate with each other on the network. The network connection might be unavailable or network settings are not configured properly.
2. Verify name resolution. One of the most common causes for replication errors is that DNS name resolution if failing. If you receive error messages indicating that the RPC server is not available or "Target account name is incorrect" errors, verify that the domain controllers can resolve the target server's FQDN.
3. Test for authentication and authorization errors. If you are receiving access denied errors during replication, then there is a problem with authentication or authorization. To identify the cause of the security error, run the dcdiag /test:CheckSecurityError /s:DomainControllerName command, where DomainControllerName is the name of the domain controller that you want to test. To test the connection between two domain controllers for replication security errors, run the dcdiag

/test:CheckSecurityError /ReplSource:SourceDomainControllerName command. This command tests the connection between the domain controller on which you run the command and the source domain controller (identified by SourceDomainControllerName). The output from these commands identifies the security issues between the domain controllers. Fix the issues and then rerun the command to verify that you have addressed the issue.

4. Check for domain controller performance issues. If a domain controller does not have enough server resources, replication may fail. For example, if the domain controller runs out hard disk space on the drive where the AD DS data store is located, the domain controller will not accept replication changes. In some cases, the domain controller performance may be the cause of delayed replication. To address domain controller performance issues, consider the following:
    - Move applications or services to another server. If the domain controller is performing multiple roles or running other applications, consider moving the roles or applications to another server on the network.
    - Distribute AD DS and DNS roles across multiple servers. AD DS integrated DNS zones provide benefits but running both AD DS services and DNS services on a single computer can cause performance issues. By distributing the load of these services, you may be able to minimize the server performance impact.
    - Deploy domain controllers with 64 bit hardware. Computers with 64 bit hardware provide significant performance gains over domain controllers with 32 bit hardware.
5. Review and modify the replication topology. In large organizations with thousands of sites, calculating the replication topology can consume the processor resources on the domain controller performing the Inter-Site Topology Generator role. Consider decreasing the number of sites in the organization or configuring dedicated bridgehead servers. As well, verify that the AD DS site link configuration corresponds with the actual WAN link configuration in your network. AD DS replication should use the WAN connections with maximum available bandwidth whenever possible.

> [Note] Two excellent resources for troubleshooting specific AD DS replication errors are the "Troubleshooting Active Directory Replication Problems" web page (*http://technet2.microsoft.com/windowsserver/en/library/4f504103-1a16-41e1-853a-c68b77bf3f7e1033.mspx?mfr=true*) and the "How to troubleshoot intra-site replication failures" web page (*http://support.microsoft.com/kb/249256*).

Tools for Troubleshooting AD DS Replication

Windows Server 2008 provides several tools for troubleshooting AD DS replication. All of these tools are installed on Windows Server 2008 when the server is configured as a domain controller.

Active Directory Sites and Services

In addition to using Active Directory Sites and Services to configure sites and replication, you can also use it perform some basic troubleshooting tasks. These tasks include:

- Forcing the KCC to recalculate the replication topology. To do this, expand the domain controller object in the AD DS site servers container, right-click NTDS Settings, point to All Tasks, and click Check Replication Topology. This forces the KCC to run immediately rather than waiting for the next scheduled update.
- Force a domain controller to pull replication changes. Locate the domain controller to which you want to pull changes in the site servers container. In the NTDS Settings container under the domain controller, right click the connection object with the domain controller from which you want to pull changes, and click Replicate now. If both domain controllers are in the same site, you will get an error message, or get a message the replication was successful. If the domain controllers are in different sites, you will get a message telling you that the domain controller will attempt immediate replication. Check the Event Viewer for replication errors.
- Force the replication of the configuration partition from or to a domain controller. When you right-click the NTDS object under a domain controller other than the domain controller that is the current focus for Active Directory Sites and Services, you can choose to Replicate configuration from the selected DC or Replicate configuration to the selected DC. One of the benefits of using these commands is that the configuration information will be replicated even if no connection object exists between the domain controllers. This option is useful when a replication partner was removed from the domain while a domain controller was offline, and the domain controller cannot locate other domain controllers to create new connection objects.

Repadmin

The most useful tool for monitoring and troubleshooting replication is Repadmin. You can use the Repadmin.exe command-line tool to view the replication topology from the perspective of each domain controller. You can also use Repadmin.exe to manually create the replication topology, force replication events between domain controllers, and view the replication metadata, and up-to-date state of vectors.

To run the repadmin command, use the following syntax:

```
repadmin command arguments [/u:[domain\]user /pw:{password|*}]
```

You need to provide the user account information only if the current logged on user is not a member of the Domain Admins group.

The following examples use some of the available command arguments for the repadmin command:

1. To display the replication partners of the domain controller named DC1, use the syntax:

```
repadmin /showreps DC1.Adatum.com
```

2. To display the highest USN on the domain controller named DC2, use the syntax:

```
repadmin /showvector dc=Adatum,dc=com  DC2.Adatum.com
```

3. To display the connection objects for the domain controller named DC1, use the syntax:

```
repadmin /showconn DC1.Adatum.com
```

4. To initiate a replication event between two replication partners

```
repadmin /replicate DC2.Adatum.com DC1.Adatum.com dc=Adatum,dc=com
```

5. Initiate a replication event for a specified directory partition with all of its replication partners

```
repadmin /syncall DC1.Adatum.com dc= Adatum,dc=com
```

Dcdiag

The Dcdiag.exe tool performs a number of tests that check domain controllers for issues that might affect replication. These tests include connectivity, replication, topology integrity, and intersite health tests.

To run the repadmin command, use the following syntax:

```
dcdiag command arguments [/v /f:LogFile /ferr:ErrLog ]
```

In the command, the optional switch /v directs the command to produce detailed output, /f directs the output to the logfile, and /ferr redirects fatal error output to a separate log file. To run all of the dcdiag tests on a local computer and display the results in the command prompt window, just type DCdiag and press ENTER. To check a remote domain controller, run DCDiag /s:Servername where Servername is the remote domain controller name.

The following are a few of the tests that can be run using DCDiag.

Connectivity

- Tests whether domain controllers are DNS registered, can be pinged, and have LDAP/RPC connectivity.

Replications

- Checks for timely replication and any replication errors between domain controllers.

NetLogons

- Checks that the appropriate logon privileges exist to allow replication to proceed.

Intersite

- Checks for failures that would prevent or temporarily hold up intersite replication and tries to predict how long it will take before the KCC is able to recover. Results of this test are often not valid, especially in atypical site or KCC configurations or at the Windows Server 2008 forest functional level.

FSMOCheck

- Checks that the domain controller can contact a KDC, a Time Server, a Preferred Time Server, a PDC, and a Global Catalog server. This test does not test any of the servers for operations master roles.

Services

- Checks whether the appropriate domain controller services are running.

Kccevent

- This test checks that the Knowledge Consistency Checker is completing without errors.

Topology

- Checks that the KCC has generated a fully connected topology for all domain controllers.

> [Note] For detailed information on how to use Repadmin and DCDiag, type the command name followed by **/?**.

Additional Tools

Two standard server administrative tools are also useful for monitoring and troubleshooting replication. The first tool is the Event Viewer. One of the event logs added to all domain controllers is a Directory Service event log. Most of the directory replication-related events are logged in this event log, and this should be one of the first places you look when replication fails.

The Reliability and Performance Monitor tool is useful for monitoring the amount of replication activity happening on the server. When a server is promoted to be a domain controller, the DirectoryServices performance object, as well as several file replication performance objects, are added to the list of performance counters. These performance counters can be used to monitor how much replication traffic there is as well as a wide variety of other AD DS–related activities.

Summary

One of the key aspects to managing Windows Server 2008 AD DS is understanding how replication works. A stable replication environment is crucial in maintaining an up-to-date copy of all directory information on all the domain controllers in the forest, which is essential to ensure consistent user logon and directory search performance. By understanding how replication works within a site and between sites, you can also design and implement the optimal replication configuration.

Best Practices

- Replication within a single site happens automatically, quickly and rarely fails. If all of your company's domain controllers are connected by fast network connections, you should implement a single site.
- On the other hand, if your company has multiple locations where you install domain controllers, creating additional sites is the easiest and best way to manage AD DS related traffic across WAN links with limited available bandwidth. Not only do multiple sites limit replication traffic, but they also keep client authentication traffic local.
- Develop a regular practice of monitoring AD DS replication. Consider using a tool such as the Active Directory Management Pack with System Center Operations Manager to

monitor replication on all domain controllers in your site. If you do not have a tool like this, regularly monitor the Directory Service event log, and either the DFS Replication event log (if your AD DS forest is at the Windows Server 2008 functional level), or the File Replication Service event log.

- In most organizations, the most important cause of AD DS replication errors is DNS lookup errors. By integrating DNS with AD DS, and taking advantage of the DNS directory partitions, you can minimize the chances of DNS errors.

Additional Resources

- "Monitoring and Maintaining Active Directory" which is Chapter 14 in this book. This chapter provides details on using monitoring tools such as Event Viewer and Reliability and Performance Monitor to monitor AD DS domain controllers, including monitoring replication..
- "Designing the Active Directory Structure", Chapter 5 in this book goes into detail on designing the AD DS site configuration.
- "Troubleshooting Active Directory Replication Problems" located at *http://technet2.microsoft.com/windowsserver/en/library/4f504103-1a16-41e1-853a-c68b77bf3f7e1033.mspx?mfr=true*. This Web site provides detailed steps for troubleshooting Active Directory replication issues and links to knowledge base articles that address specific Event IDs.
- The "How to troubleshoot intra-site replication failures" knowledge base article at *http://support.microsoft.com/kb/249256* provides details on how to troubleshoot intra-site replication errors. This KB article, as well as many of the other KB articles listed next, refer to Windows Server 2003. Many of the steps in troubleshooting AD DS replication have not changed in Windows Server 2008
- The "Active Directory Replication Troubleshooter" located at *http://blogs.technet.com/rbeard47/pages/active-directory-replication-troubleshooter.aspx* provides a detailed step by step process for troubleshooting Active Directory replication.
- "Fixing Replication DNS Lookup Problems (Event IDs 1925, 2087, 2088)". *http://technet2.microsoft.com/windowsserver/en/library/43e6f617-fb49-4bb4-8561-53310219f9971033.mspx?mfr=true*
- "How to troubleshoot RPC Endpoint Mapper errors". (*http://support.microsoft.com/kb/839880*)
- "Service overview and network port requirements for the Windows Server system" *(http://support.microsoft.com/kb/832017*). This article describes the ports required by most Windows Server services, including AD DS replication. This information is very useful when configuring firewalls between domain controllers.
- "Replication Not Working Properly Between Domain Controllers After Deleting One from Sites and Services" *(http://support.microsoft.com/kb/262561*)
- "Active Directory Replication Technologies" *http://technet2.microsoft.com/windowsserver/en/library/74d58697-970a-45db-9139-ebcd3db051181033.mspx?mfr=true*

- The "Script Repository: Active Directory" web site, located at *http://www.microsoft.com/technet/scriptcenter/scripts/default.mspx?mfr=true* has several scripts that can be used to enumerate and modify the AD DS site and site link configuration.

Related Tools

Windows Server 2008 provides several tools that can be used when managing and troubleshooting replication. Table 4-2 lists some of these tools and when you would use each of the tools.

Table 4-2 AD DS Replication Tools

Tool name	Description and usage
Dnslint.exe	This tool is a free download from Microsoft. (See *http://support.microsoft.com/kb/321045* for the download location). This tool can be used to help diagnose common DNS name resolution issues and to verify that DNS records used specifically for AD DS replication are correct.
Nslookup.exe	This tool is included in all Microsoft Windows server and client operating systems. Nslookup is used to query DNS servers and to obtain detailed responses. The information obtained using Nslookup can be used to diagnose and solve name resolution problems, verify that resource records are added or updated correctly in a zone, and debug other server-related problems.
Active Directory Sites and Services	Use to configure sites and replication and to perform some basic AD DS replication troubleshooting tasks.
Repadmin.exe	Use this command-line tool to view the replication topology from the perspective of each domain controller. You can also use Repadmin.exe to manually create the replication topology, force replication events between domain controllers, and view the replication metadata, and up-to-date state of vectors.
DCDiag.exe	Use this tool to perform tests that check domain controllers for issues that might affect replication

Resources on the CD

- "ADDSSite.xlsx" is a spreadsheet template for documenting AD DS site information.
- "ListADDSSites.ps1" is a simple Windows PowerShell script for listing information about all of the sites in your forest.

Related Help Topics

- "Checklist: Configure an Additional Site" in Active Directory Sites and Services help.
- "Checklist: Configure the Intersite Replication Schedule" in Active Directory Sites and Services help.

  "Troubleshooting Active Directory Domain Services Replication" in Active Directory Sites and Services help.

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	• **Account management** - includes creating, maintaining, and removing user accounts. • **Security group management** - includes creating security groups, provisioning security groups to grant access to network resources, managing memberships of security groups, and removing security groups. • **Resource management** - includes all aspects of managing network resources such as end-user workstations, servers, and resources hosted on servers such as file shares or applications. • **Group Policy management** - includes all aspects of creating, assigning, and removing Group Policy objects within the Active Directory structure. • **Application-specific data management** - includes all aspects of managing Active Directory-integrated or -enabled applications such as Microsoft Exchange Server.
Service management	• **Installation and trust management** - includes aspects such as the creation and deletion of domains, the deployment of domain controllers, and the configuration of appropriate Active Directory functional levels. • **Domain controller and directory database management** - includes aspects related to the management of domain controller hardware, database maintenance, and the application of service packs and security updates. • **Schema management** - includes the extension or modification of the schema to support the deployment of Active Directory-enabled applications. • **Operations master roles management** - includes tasks that ensure the proper assignment and configuration of operations master roles.

- **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.

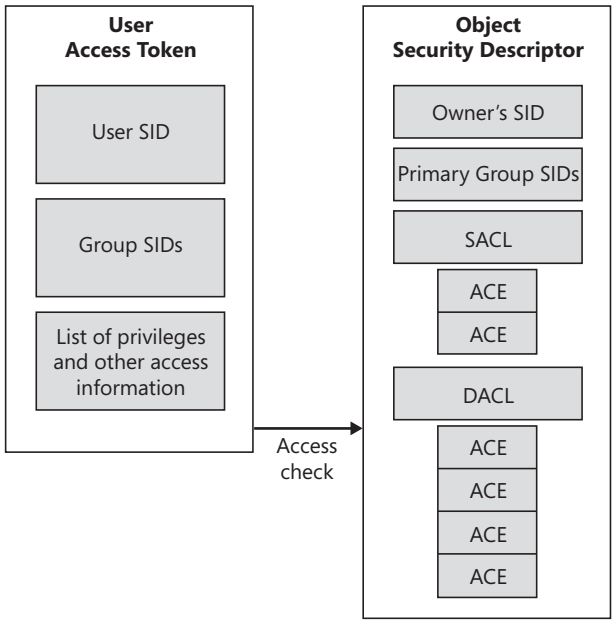


Figure 9-1 Access check between a user's access token and an object's security descriptor.

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.

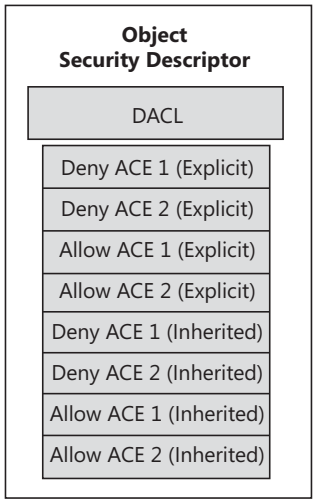


Figure 9-2 Evaluating Allow and Deny ACEs.

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.

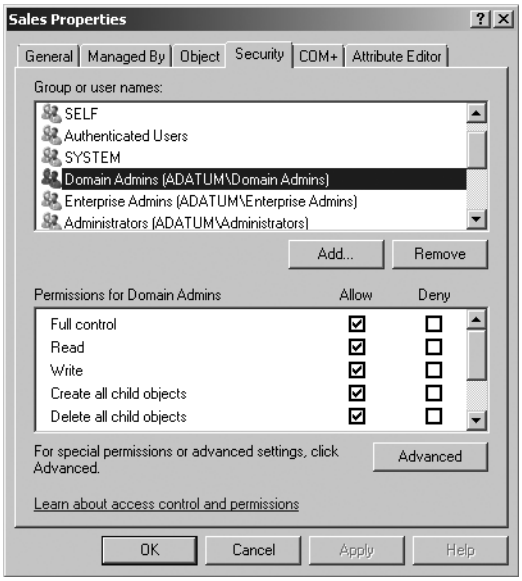


Figure 9-3 Viewing the Security page on an Organizational Unit object.

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.

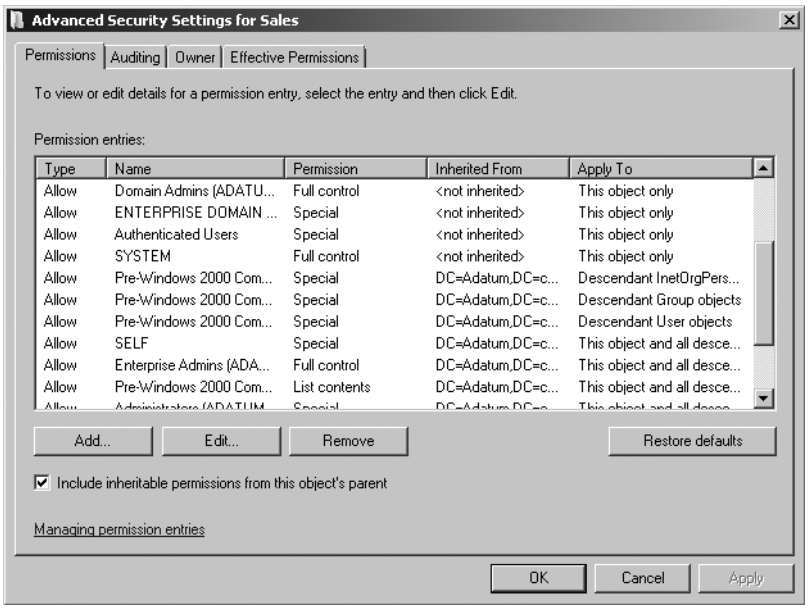


Figure 9-4 Viewing the Advanced Security Settings for an object.

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.

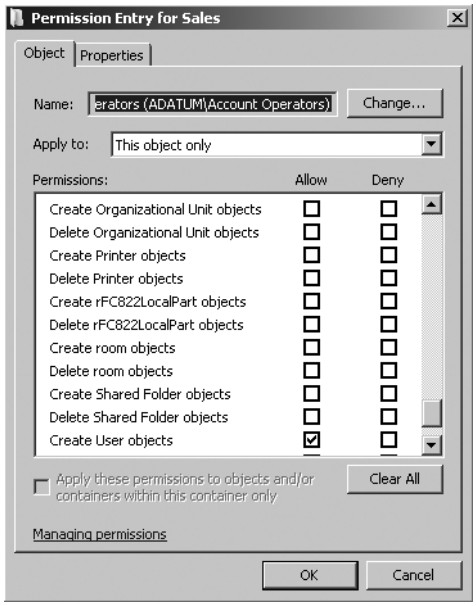


Figure 9-5 Assigning special permissions to Active Directory objects.

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.

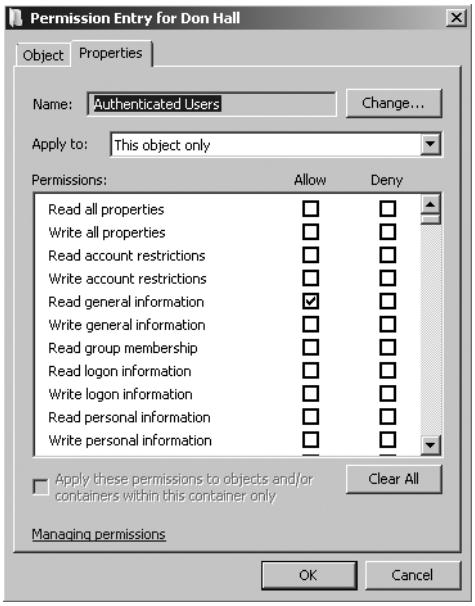


Figure 9-6 Configuring an object's property permissions.

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.

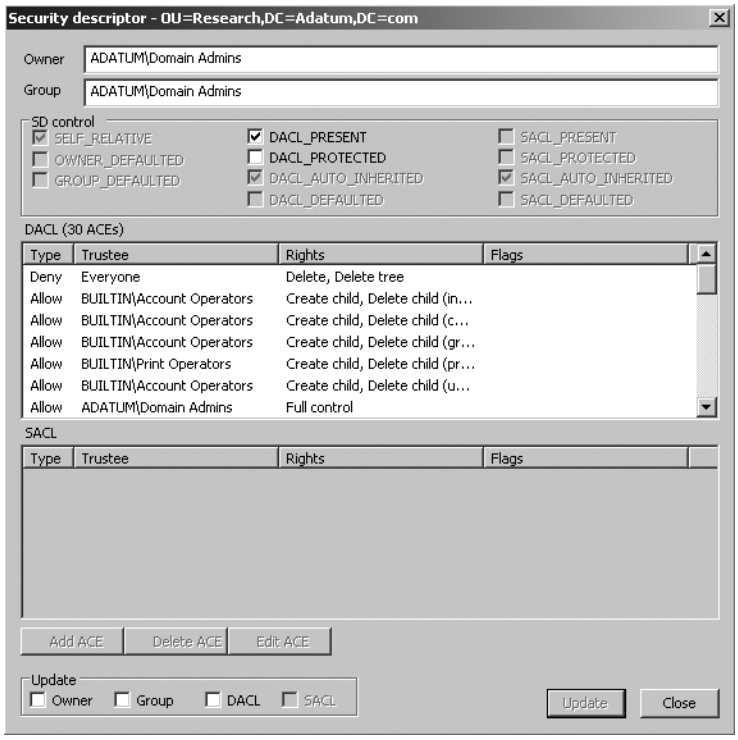


Figure 9-7 Using Ldp.exe to modify the security descriptor.

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.

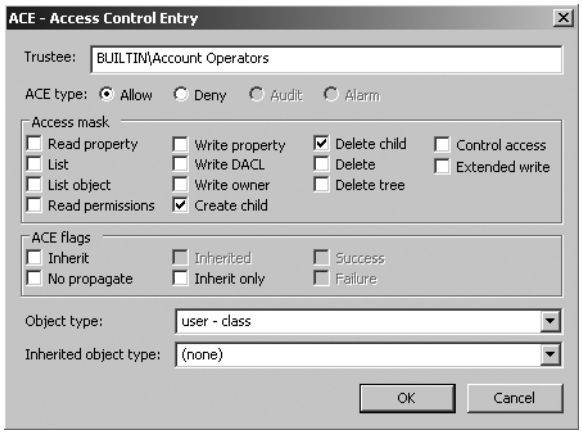


Figure 9-8 Modifying an ACE using Ldp.exe.

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.

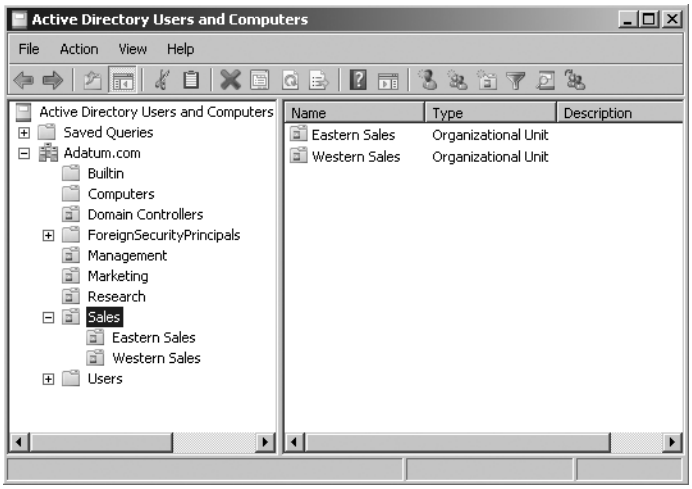


Figure 9-9 Delegating management of the Sales OU.

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.

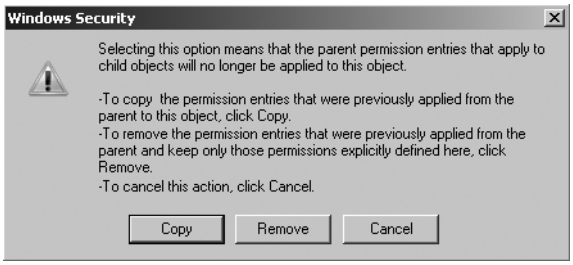


Figure 9-10 Selecting the option to copy or remove permissions when blocking permissions inheritance.

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.

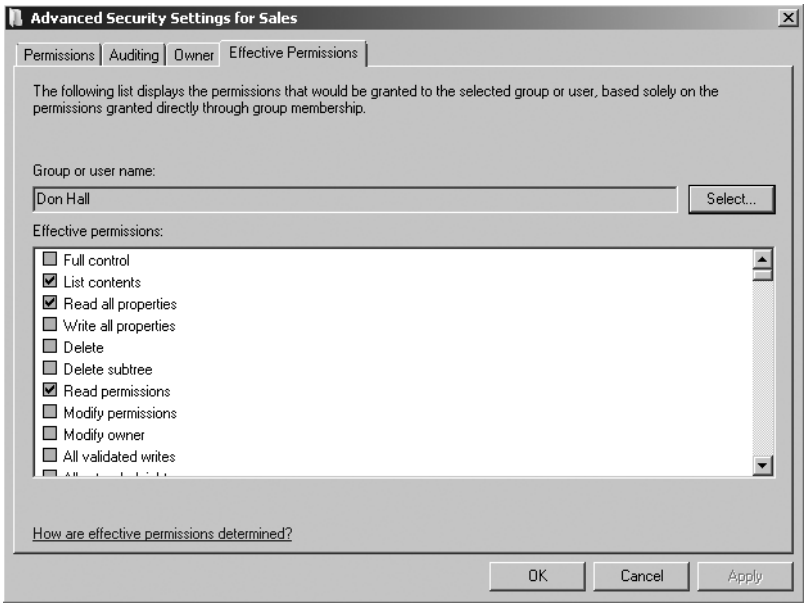


Figure 9-11 Displaying the effective permissions for an Active Directory object.

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.

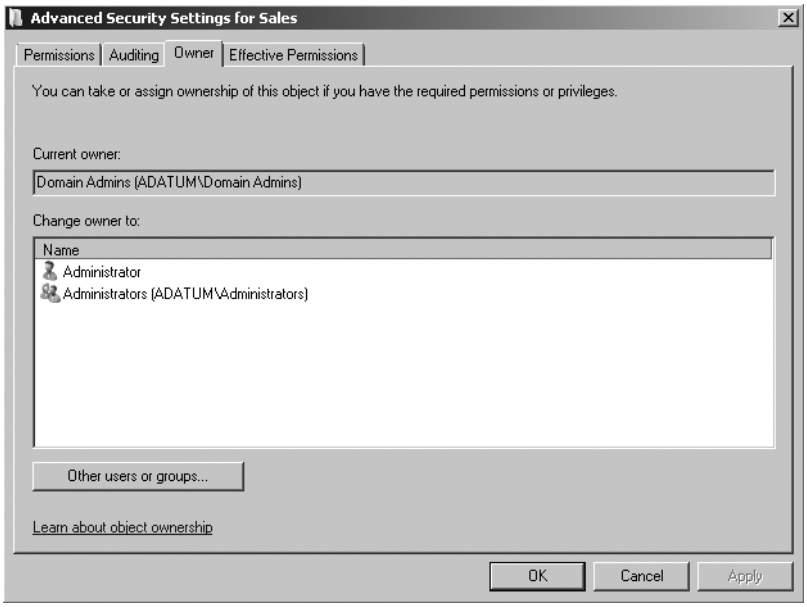


Figure 9-12 Viewing the ownership of an Active Directory object.

If you have the Modify owner permission to the object, you can use this interface to modify the owner of the object. You can chose 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.

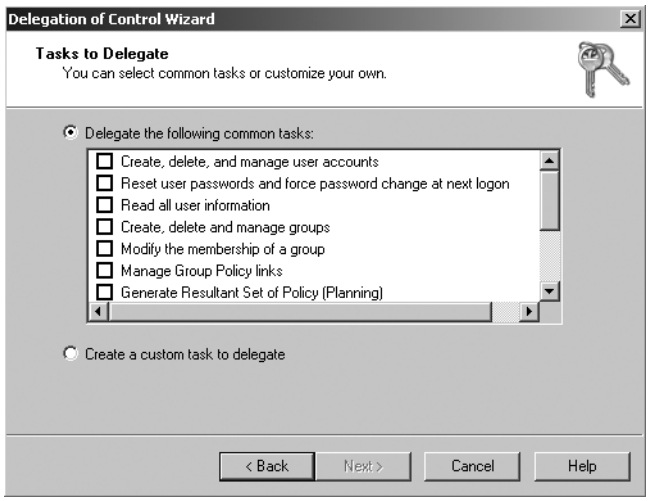


Figure 9-13 Using the Delegation of Control Wizard to select a common task or create a custom task to delegate.

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.)

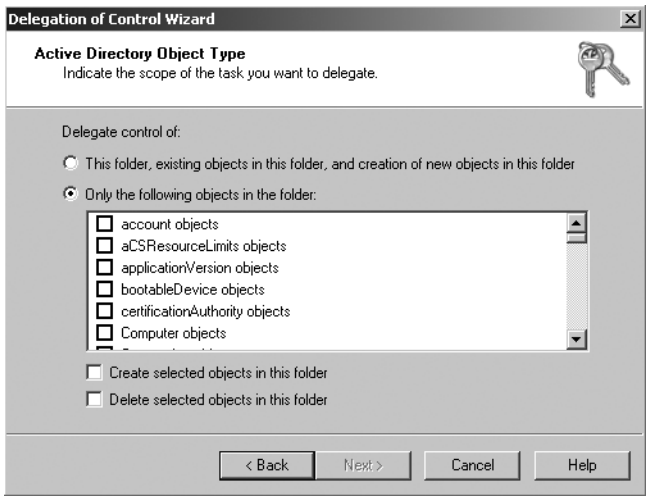


Figure 9-14 Selecting the type of object or objects to which permissions will be delegated.

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.)

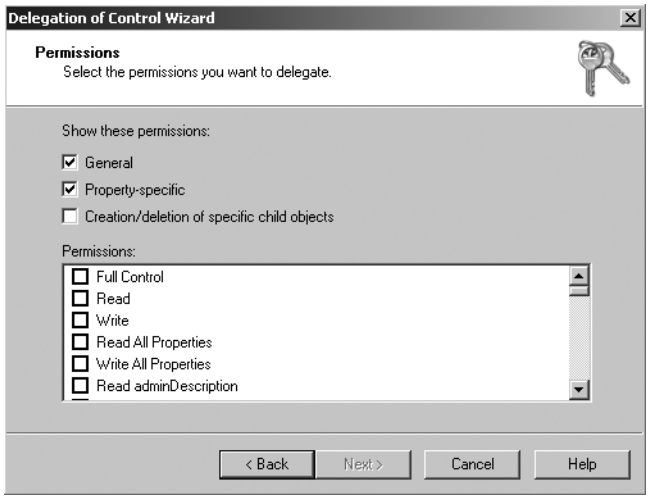


Figure 9-15 Selecting the specific permissions to delegate.

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.

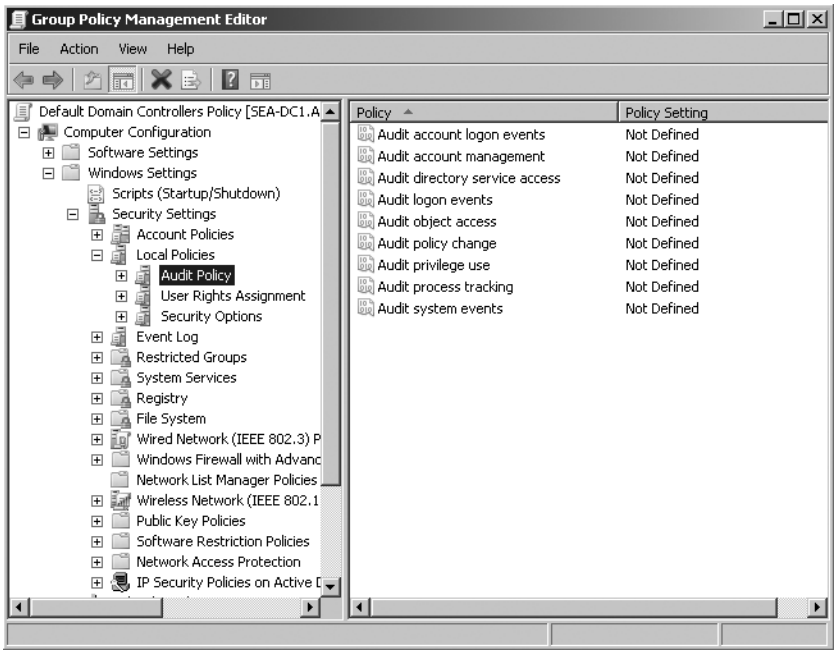


Figure 9-16 Configuring auditing on the Default Domain Controllers OU.

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.

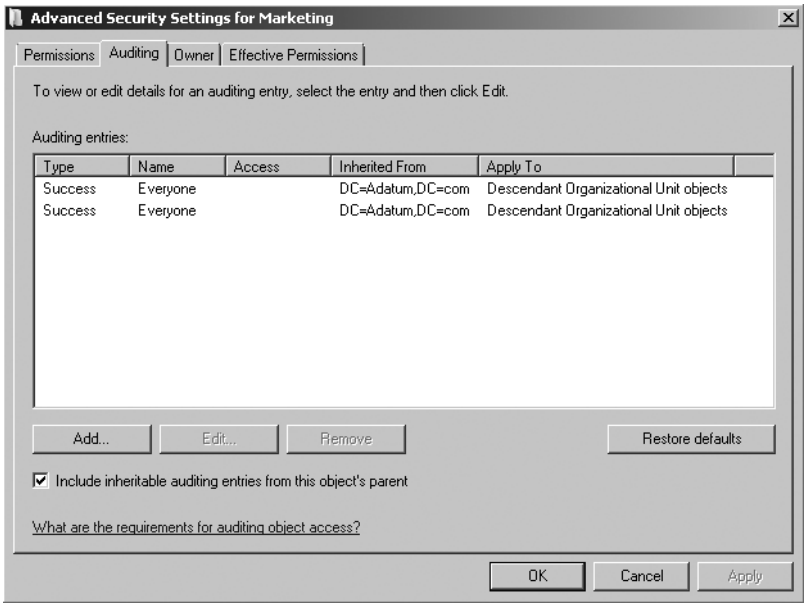


Figure 9-17 Configuring auditing on Active Directory objects.

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.mspx*. 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.

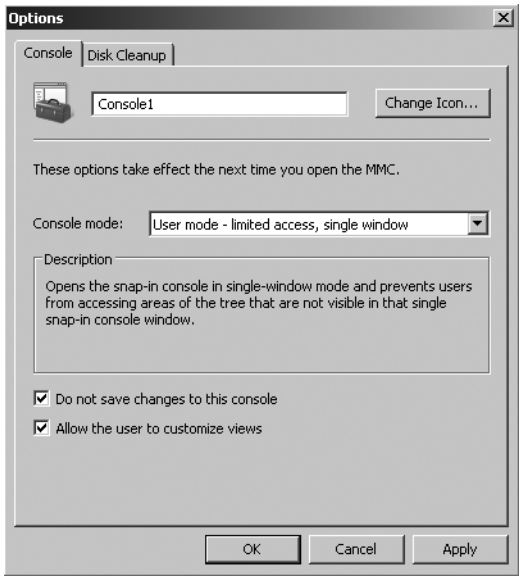


Figure 9-18 Configuring a custom MMC to prevent changes to the MMC.

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>