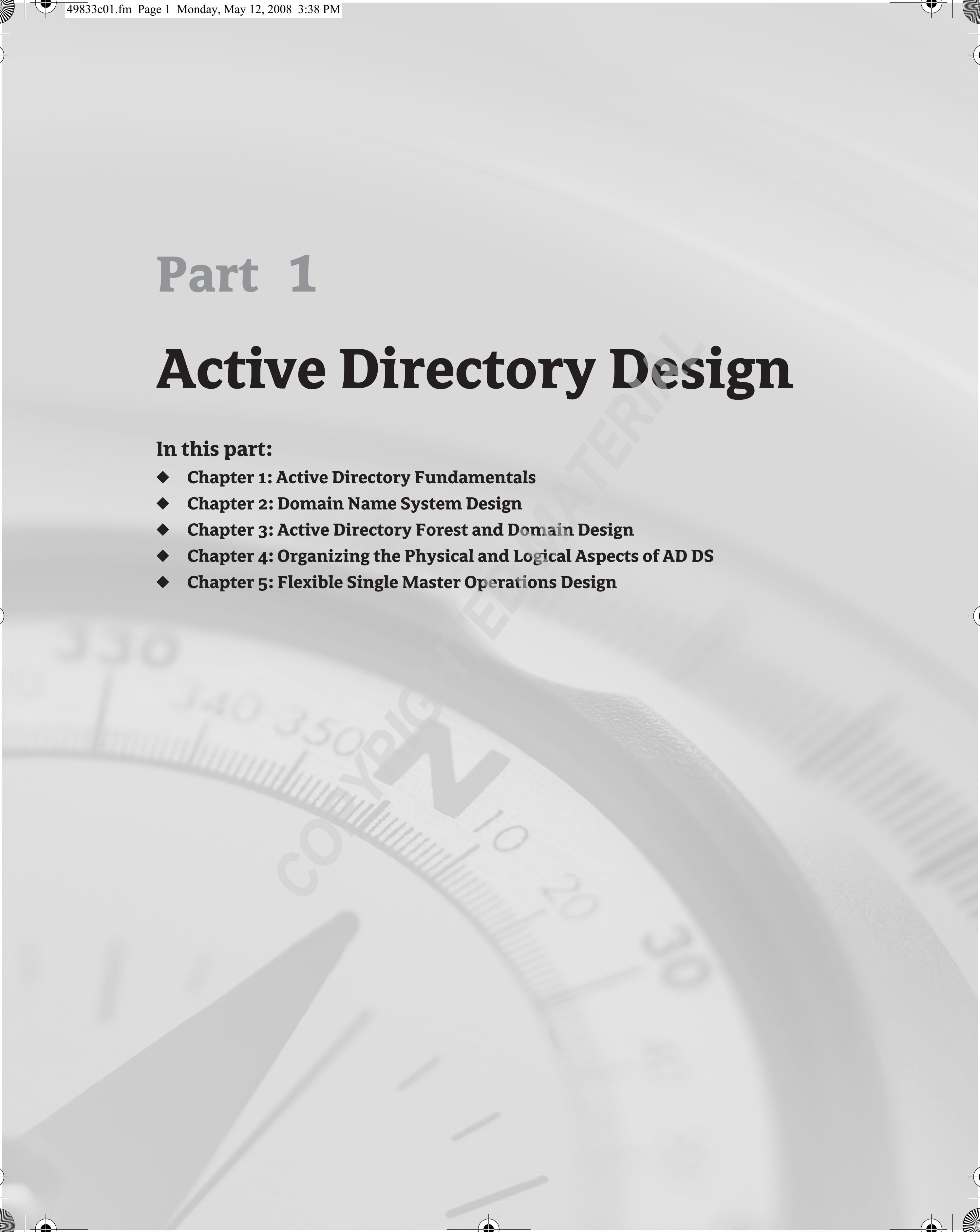


Part 1

Active Directory Design

In this part:

Chapter 1

Active Directory Fundamentals

Since the inception of network operating systems, the men and women who are responsible for administering and managing them have wanted an easy way to do so. Networks have gone through a natural evolution from peer-to-peer networks to directory-based networks. Directory-based networks have become the preferred type of network because they can ease an administrator's workload.

To address the needs of organizations, the Institute of Electrical and Electronics Engineers (IEEE) developed a set of recommendations that defined how a directory service should address the needs of administrators and efficiently allow management of network resources. These recommendations, known as the X.500 recommendations, were originally envisioned to include a large centralized directory that would encompass the entire world, divided by geopolitical boundaries. Even though X.500 was written to handle a very large amount of data, designers reviewing the drafts of these recommendations saw merit in the directory and soon the recommendations were adopted by several companies, including the two best known, Novell and Microsoft.

Active Directory is Microsoft's version of the X.500 recommendations. Battles rage between directory services camps, each one touting its directory service as the most efficient one. Because some of the directory services, such as Novell Directory Services (NDS) and eDirectory, have been around longer than Active Directory, those that are familiar with NDS will attack Active Directory. Their attacks are usually focused on the idea that Active Directory does not perform functions the same way that NDS does.

When it is all said and done, companies that develop X.500-based directory services can interpret the recommendations and implement them to fit their design needs. Microsoft interpreted and employed the X.500 recommendations to effectively manage a Windows-based network. Novell did the same for a Novell-based network, and the two for years have been at odds over which is more efficient. All that notwithstanding, Microsoft has enjoyed great success with Active Directory. It has been adopted by thousands of organizations and will more than likely continue to be used for many years to come.

Do I Need Active Directory?

Active Directory is the database (think of a directory as a collection of information, like a phone book), whereas a domain controller is a single computer or server that controls Active Directory. There are typically multiple domain controllers that host Active Directory.

How do you know if you need Active Directory? There are factors that you should address to determine whether you should defer installation of a domain controller. Following are some of the questions you should ask:

Do I want to centrally manage access to resources such as printers, users, and groups?

Do I want to control user accounts from one location?

Do I have applications that rely on Active Directory?

If you answered "yes" to any of these questions, you undoubtedly will want to take advantage of the features that Active Directory provides. Taking each one of the questions into account, you will find that your life as an administrator will be much easier if you use Active Directory over using no directory service whatsoever. The tools that become available when you implement Active Directory will ease your administrative load, although there is an inherent learning curve associated with any new technology.

If you answered "yes" to the last of the three questions just posed, you have no choice but to implement Active Directory. Most of the Active Directory–enabled applications on the market rely on the installation of a full version of Active Directory within your network. There are some Active Directory–enabled applications that can take advantage of using Active Directory Lightweight Directory Services (AD LDS) –based systems. AD LDS is discussed later in this chapter.

The first two questions relate to something for which administrators have strived over the years. Having one central location to manage users and resources makes an administrator's life easier. If you have to continually move from server to server to administer the resources contained on them, you will spend more time tracking down the resources than you would performing your job. If you have to maintain user accounts on several systems, you must make sure you have an efficient method of cataloging the accounts so that you know where they reside.

With Windows 2000 Server, Windows Server 2003, and now Windows Server 2008, you can use Active Directory Domain Services (AD DS) as the central repository for user, group, and computer accounts as well as for shared folders and printers. Having the ability to manage these resources from any domain controller within your domain allows you to greatly reduce your administrative overhead.

The Basics

When you break it down, Active Directory is a type of database, but one built as a "directory." The difference between a relational database and a directory is that the former is optimized for updating, while the latter is optimized for reading. In this manner, Active Directory was developed with the understanding that the objects contained within the directory would not be changing often, but would be used for users, computers and administrators to control, manage, and discover the organization's resources.

One of Active Directory's most basic functions is that it provides a centralized repository for user account information. When an administrator creates a user account, the account information is held on a domain controller within the domain in which the user resides. All of the domain controllers within the domain will receive an identical copy of the user account so that the user is able to authenticate using any domain controller in the domain.

Any changes to the user account are made on one of the domain controllers and then sent to every other domain controller within the domain. This transfer of data is called *replication*. Replication of information can be a burden on the network, especially in environments with several thousand users, groups, computers, and other objects. To alleviate the replication burden on the network, Active Directory replicates only the attributes that have been changed, and not the entire object.

To get a good understanding of how Active Directory works, you must first understand what the schema is and the role it plays in the directory service. The following section will outline the major roles of the schema.

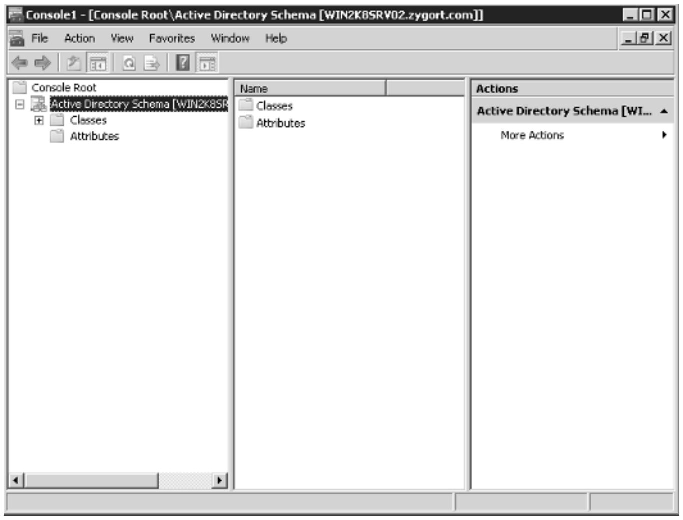
Schema

The *schema* (i.e., a structured framework or plan) acts as the building blocks of Active Directory, much like DNA molecules are the building blocks for our bodies. Just as our DNA holds all of the information necessary to build our leg, ears, hair, ear hair, etc., the schema holds all of the information needed to create users, groups, computers, and so on within Active Directory. The schema defines how each attribute can be used and the properties associated with the attribute. Take, for instance, a child's toy that we have grown up with: LEGOs. When you first take a look at LEGO bricks, you see hundreds of tiny pieces that really don't seem to represent anything. Some are short, some are long, and some are special shapes. These are the individual pieces, or building blocks, that will go into creating the buildings, cars, airplanes, and dioramas.

The Active Directory schema is pretty much the same thing. If you look within the Active Directory Schema snap-in you will see hundreds of entries that are used when creating objects within Active Directory. As you expand the Active Directory Schema section of the tool, shown in Figure 1.1, you will see the window that contains classes and attributes. The entries known as attributes allow you to create new objects or modify existing objects within your directory.

To add the Active Directory Schema snap-in to a Microsoft Management Console (MMC), you will first need to register the dynamic link library. To do so, open the Run line or use a command prompt on the domain controller and type in `regsvr32 schmmgmt.dll`.

FIGURE 1.1
Active Directory Schema snap-in



Console1 - [Console Root\Active Directory Schema [WIN2K8SRV02.zygort.com]]

ATTRIBUTES

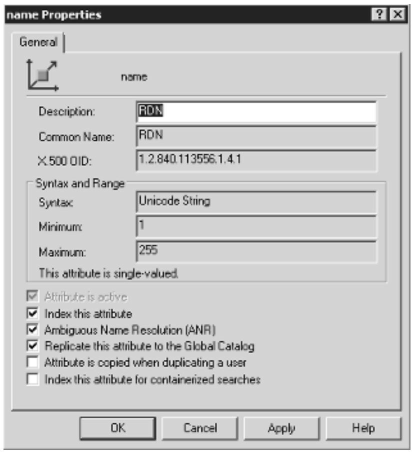
To standardize Active Directory, the schema defines the attributes that can be used when creating objects. Unlike our LEGO bricks, however, these attributes are defined only once and can be used for any object. Defining the attribute once and using it for multiple objects allows for a standardized approach of defining objects, especially when searching for the attribute. Take the `name` attribute, for example; whenever an object uses the `name` attribute you know that the name has to be at least one character in length and cannot exceed 255 characters. You would know this because of the syntax and rules that are applied to the attribute.

The Properties page of the `name` attribute is shown in Figure 1.2. There is a lot of information within this page, but right now we are interested only in the Syntax and Range area. Notice that the attribute is a Unicode string that has to be at least one character in length and cannot exceed 255 characters. Each attribute within the schema is defined in such a manner, although the syntax for each of the attributes could be different.

The properties for `Bad-Pwd-Count` are shown In Figure 1.3. This is another attribute that makes up a user object. Notice that the X.500 Object Identifier (OID) is different from that of the `name` attribute. Each attribute within the schema has to have a unique OID. These are registered and maintained by the Internet Assigned Numbers Authority (IANA). Once assigned, the OID should not be used by any other attribute. Within Active Directory, the default attributes are already assigned OIDs, and those OIDs are protected in a way that will not allow another application to overwrite them.

New attributes will need to be assigned an OID. If you are adding an attribute for use in an object, you should register it with the IANA to safeguard the attribute and to make sure that it does not step on any other attributes. Registration is free, and as long as your OID is unique, you should be issued an OID for your attribute. The attributes that Microsoft uses are all within their own OID range, which starts with 1.2.840.113556. For a complete list of the registered OIDs, visit `http://asn1.elibel.tm.fr/oid/index.htm` and perform a search on the OID. If you have registered an OID, it will appear in this database once the entry is added.

FIGURE 1.2
Name attribute property page



name

Description: RDN

Common Name: RDN

X.500 OID: 1.2.840.113556.1.4.1

Syntax and Range

Syntax: Unicode String

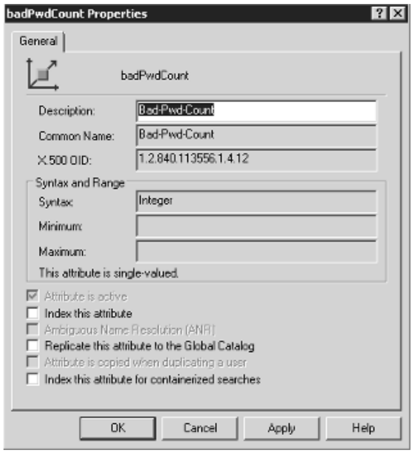
Minimum: 1

Maximum: 255

This attribute is single-valued.

☑ Attribute is active
☑ Index this attribute
☑ Ambiguous Name Resolution (ANR)
☑ Replicate this attribute to the Global Catalog
☐ Attribute is copied when duplicating a user
☐ Index this attribute for containerized searches

FIGURE 1.3
Bad-Pwd-Count attribute property page



badPwdCount

Description: Bad-Pwd-Count

Common Name: Bad-Pwd-Count

X.500 OID: 1.2.840.113556.1.4.12

Syntax and Range

Syntax: Integer

Minimum:

Maximum:

This attribute is single-valued.

☑ Attribute is active
☐ Index this attribute
☐ Ambiguous Name Resolution (ANR)
☐ Replicate this attribute to the Global Catalog
☐ Attribute is copied when duplicating a user
☐ Index this attribute for containerized searches

Within an attribute's properties, you will find several check boxes that you can select. Each of them is described in the following list:

Attribute Is Active You can deactivate attributes that you no longer need within Active Directory. Note that the default attributes cannot be deactivated, nor can attributes that are still in use within an object.

Index This Attribute If this is an attribute on which you are going to allow searches, you may want to index the attribute to increase the search responsiveness.

Ambiguous Name Resolution (ANR) When you select this option, you allow a Lightweight Directory Access Protocol (LDAP)–based client to resolve a request when only partial data is available.

Replicate This Attribute to the Global Catalog Not every attribute needs to reside within the global catalog. The rule of thumb is, if you need to locate an object based on an attribute or if the object's attribute is needed within another domain, you should add it. Otherwise, to reduce the total size of the domain partition you should not add in any superfluous attributes.

Attribute Is Copied When Duplicating a User When you copy a user account, several attributes are copied from the original account to the new account. If you want the attribute to copy, select the box. Do note that many attributes are unique to a user, so select this option with care.

Index This Attribute for Containerized Searches If you select this option, the attribute can be indexed for searches within containers, such as organizational units (OUs), in Active Directory.

Object Classes

An object class is a defined grouping of attributes that make up a unique resource type. One of the most common object classes is the user class. Use the user object class as the template for a user account. When you create a user account, the attributes that are defined for the user object class are used to define the new account. Information that you populate within the Add User wizard or enter within the `dsadd` command line become the properties within the attributes.

If we go back for a minute to the LEGO metaphor, you can use some of the brown blocks available to create a roof on a house, some red bricks to make the walls, and tan bricks to make a door. The clear pieces can be used as windows and the white pieces form the porch. Each of these individual items (the bricks, the color of the bricks, the shape of the bricks, and the placement of the bricks) is considered an attribute. Putting these attributes together forms the object class "house." When you build your first house, you have built your first object. Subsequent houses will have the same attributes, but you may build the porch with tan pieces instead of white ones.

So, when I create a user account for Maria, that user account will have unique values stored within the attributes for her user account. Bob's user account will be created using identical attributes, but will not have the same values within each attribute. Maria's phone number may be 555.1234, and Bob's 555.9876.

Not all of the attributes that make up an object class are shown within the administrative tools. Many of them hide behind the scenes and will rarely, if ever, need to be changed. One such attribute is the user's Security Identifier or SID. The user's SID will change when a user is moved from one domain to another, but will not change while the user remains within a domain. The Active Directory Users and Computers management tool does not have the ability to change this attribute. A default set of attribute fields appears within the utilities, and if you decide to make an attribute available for updating, you may need to programmatically add the fields to the utilities.

Attributes are defined as mandatory or optional. Mandatory attributes have to be populated, or the object will not be created. One such attribute is a computer's name. Optional attributes do not necessarily need to have values. Attributes such as Manager within a user object does not need to be populated, but it is always nice to include that information. The more complete the information, the more useful Active Directory becomes.

The Two Sides of AD

Active Directory has both a logical side and a physical side, and each one plays a very important role. The physical side is made up of the domain controllers and physical locations where the domain controllers reside. When you promote a system to domain controller status, you will usually place that domain controller close to the user population that will use it for authentication and access. Domain controllers need to communicate with one another to share the information they have.

The logical side is a little more nebulous; as well as containing the objects that define how the resources are organized and accessed, the logical side contains objects within Active Directory that define how the domain controllers will communicate with one another. Active Directory sites and site links define which domain controllers will replicate directly with each other and which ones will have to communicate indirectly through other domain controllers.

Domains dictate the replication scope. When you create a domain, the domain partition is replicated only to domain controllers from the same domain. The domain partition is not copied to domain controllers outside of the domain. This allows you to partition your directory service and reduce the size of the database file that holds all of the forest's objects. Forests and domains are discussed in greater detail in Chapter 3, "Active Directory Forest and Domain Design."

Organizational units are used to organize objects for easy administration and to manage those objects easily using group policies. To have efficient administration of resources, you should design your Active Directory with administration in mind.

The design of the logical and physical sides of Active Directory is discussed in great detail in Chapter 4, "Organizing the Physical and Logical Aspects of Active Directory." If you are in the process of rolling out Active Directory, be sure to develop a detailed plan for the rollout. Without a good design, Active Directory may not work efficiently for your environment. If your design does not meet the needs of your organization, you may be faced with either suffering through working with an inadequate design or rebuilding your Active Directory infrastructure from the ground up. Neither of these options will sit well with your user base or the management of the company.

What's New in Windows Server 2008?

Windows Server 2003 shipped in the spring of 2003. When released, it was the most advanced network operating system Microsoft had ever developed. The advances that it made over Windows 2000 Server were obvious almost immediately; even though most of the new functionality was seen only by administrators, Microsoft went to great lengths not only to enhance the security and functionality in Windows Server 2003, but also to include additional administrative tools to make an administrator's life easier. And if you know anything about administrators, you know that anything that makes their life easier, they like.

Over the course of the two and a half years from the time Windows Server 2003 shipped and Windows Server 2003 R2 became available, several new technologies developed that Microsoft wanted to take advantage. Also, new codes with patches for new attack vectors that posed security risks to an organization's resources had been developed and needed inclusion.

Several enhancements to Active Directory were included with the R2 release. Those enhancements include Active Directory Application Mode (ADAM)—now known as Active Directory

Lightweight Directory Services (AD LDS), Active Directory Federation Services (AD FS), and Unix Identity Management.

Windows Server 2008 builds on these technologies and incorporates other stand-alone Microsoft products to become the most robust operating system Microsoft has released to date.

What's in a Name?

With the release of Windows Server 2008 and the inclusion of several enhancements to AD, Microsoft has decided to realign all of its "identity" technologies under the Active Directory umbrella. Some items have simply been renamed; other technologies have been moved into the Active Directory Family. With all of these changes, and in typical Microsoft fashion, there are some new names to get familiar with. (These new technologies are discussed in subsequent subsections.)

- The Active Directory that we've all grown to know and love is now known as Active Directory Domain Services (AD DS). AD DS stores all information about resources on the network, such as users, computers, and other devices.
- Active Directory Lightweight Directory Services (AD LDS) is the latest version of Active Directory Application Mode (ADAM).
- Active Directory Federation Services (AD FS) provides Web single sign-on (SSO) technologies to authenticate users to multiple web applications in a single session.
- Active Directory Rights Management Services (AD RMS) is an information-protection technology that works with RMS-enabled applications to protect and secure information from unauthorized use online and offline, inside and outside of the environment.
- Active Directory Certificate Services (AD CS) allows the mapping of users and resources to a private key to help secure identity in a Public Key Infrastructure (PKI)-based environment.

Along with renaming and restructuring these technologies, Microsoft (MS) also updated all of the existing Active Directory technologies. Following are some of the major updates to Active Directory:

- Read -only domain controllers (RODCs) allow organizations to easily deploy a domain controller in locations where physical security cannot be guaranteed.
- Windows Server Core has introduced a new edition of Windows Server titled "Server Core". Server Core is a Windows 2008 server that is command line–driven and does not possess a GUI.

Active Directory Lightweight Directory Services

Active Directory Lightweight Directory Services (AD LDS) allows administrators to create small versions of Active Directory that run as non–operating system services. Because AD LDS does not run as an operating system service, it does not require deployment on a domain controller. Any workstation or server can host an instance, or multiple instances, of AD LDS. Instead of building a domain controller so that developers have an Active Directory database to work with, you could create an instance of AD LDS on their workstations for them to test against. You could also use it as a repository for data used by a customer-relations management program or an address book directory. If you need a directory to hold data instead of a database, you may want to consider using AD LDS.

One of the biggest benefits of using AD LDS is its administrative benefits. Because AD LDS is a user version of Active Directory, anyone familiar with how to manage objects within Active Directory should be at ease when working with objects in AD LDS. And as in Active Directory, you can control your replication scope and the systems with which you replicate objects. If you have three systems that need to host the directory, you can specify that the AD LDS partitions be hosted on those systems.

Until the release of Exchange 2007, developers were more interested in AD LDS than were most administrators. For developers, the possibilities provided by AD LDS are limited only by imagination. If an application's primary use of data is reading that data and performing queries against that data rather than making mass changes, AD LDS should fit the bill.

Exchange 2007 introduced a new Exchange server role, the Edge Transport role. An Edge Transport server is not a member of your Active Directory domain and usually sits in your demilitarized zone (DMZ). Among other functions of the Edge Transport role, you can configure AD LDS in the DMZ to help facilitate the Active Directory account lookups.

For more information on AD LDS and how to manage it within your infrastructure, turn to Chapter 14, "Maintaining the Active Directory Database."

Active Directory Federation Services

Many organizations are partnering with businesses to efficiently deliver products and services. As businesses form these alliances, there needs to be a secure method of authenticating users from the partners' organizations. Part of the challenge to allowing authentication into your network is the security needed to maintain the connection between partners while keeping hostile entities at bay. In the past, this was possible with several tools and utilities, none of which appeared to work well with each other.

Active Directory Federation Services (AD FS) extends Active Directory to the Internet while guaranteeing the authenticity of the accounts attempting to authenticate. Using this technology will not only enable organizations to work with partner organizations more efficiently; it will also allow interoperability with a with range of applications and platforms, such as Netegrity, Oblix, and RSA, as well as leverage client systems that can utilize Simple Object Access Protocol (SOAP)–based command sets.

When using AD FS, an organization can allow users that exist within separate forests, as well as among partner organizations, to have access to the organization's web applications and use a single sign-on. AD FS is based on the Web Services (WS-*) architecture that is being developed with the cooperation of several companies, including IBM and Microsoft. Chapter 10, "Managing Access with Active Directory Services," will cover managing and maintaining AD FS integration between organizations and within an organization.

Active Directory Rights Management Services

Microsoft released Windows Rights Management Services (RMS) a few years ago. Windows Server 2008 introduces a pretty significant update to this product and has changed the name to Active Directory Rights Management Services (AD RMS).

Chapter 11, "Managing Active Directory Rights Management Services," details AD RMS and all the new features that have been introduced in Server 2008. Previously available as a separate download, AD RMS is now a feature of Active Directory and has been included in the base product.

Active Directory Certificate Services

The Active Directory Certificate Services (AD CS) allow you to create and manage certificates used in environments that employ public-key technologies. AD CS allows you to associate the identity of a person, device, or service to a private key.

AD CS is not a new technology, but it is new to the Active Directory family. This book will dive deep into Certificate Services, as well as highlight the changes that are included in Server 2008.

One of the biggest changes is the addition of Cryptography API: Next Generation (CNG). CNG allows administrators to use custom algorithms with Active Directory, with Secure Sockets Layer (SSL), and with Internet Protocol Security (IPSec). This is accomplished by using the U.S. government's Suite B cryptographic algorithms.

Enhancements such as Online Certificate Status Protocol support, Network Device Enrollment Service, web enrollment, restricted enrollment agent, and PKIView will be discussed in greater detail in Chapter 12, "Managing Active Directory Certificate Services."

Windows Server Core

In keeping with Microsoft's ongoing battle against all things security (whether implied or true), the company has introduced a new type of server for 2008. Windows 2008 Server Core is a Windows server that does not contain a GUI. All administration of Server Core is performed via the command line or via scripting. You may also administer some functions by connecting to Server Core from another server's Microsoft Management Console (MMC) utility.

Server Core was introduced for many reasons:

- Reduced maintenance—Server Core installs only what is necessary for the specific server role.
- Reduced attack surface—Because Server Core installs only what is necessary for the specific server role, fewer applications are running on the server, and the attack surface is reduced.
- Reduced management—Because fewer applications are running on the server, there is less to manage. (Noticing a trend here?)
- Less disk space—Server Core can run on less that 5 GB of disk space. Considering that most new servers come standard with 150-plus GB drives now, you may be wondering why this is an advantage of Core Server. Think about what is being done with solid-state drives in the marketplace right now. There may be options for running Server Core on solid-state drives in the very near future.

Read-Only Domain Controller

With the release of Windows Server 2008, Microsoft has introduced the read-only domain controller (RODC). The RODC contains a read-only copy of the Active Directory database that cannot be directly configured. This increases security, especially in areas where the physical security of the domain controller cannot be guaranteed.

A new Domain Name System (DNS) zone was also created to support this new server type. A primary read-only zone contains read-only copies of the domain partition, ForestDNSZones, and DomainDNSZones. More information about the changes to DNS in Windows Server 2008 can be found in Chapter 2, "Domain Name System Design."

Server Manager

At first glance, Server Manager seems to be just another attempt by Microsoft to put some things together that they think would be in our ideal tool chest. I dismissed it in the beginning, but after working with it for a while, it has become one of my favorite new features of Server 2008.

Real World Scenario

RODC in Action

Carlos is an administrator of a small bank with five branch offices. Because of the regulations that banks have to follow, Carlos cannot deploy a domain controller at a remote site unless he can guarantee physical security of the server.

Each of the five branch offices has one room that contains the shared printers, copiers, and all of the office supplies. Carlos decides that this room is the only room he can place the domain controllers.

Carlos installs five new Windows 2008 RODCs on his network, one in each remote branch. This allows Carlos to place domain controllers at the remote site in an unsecured area, and users at the remote sites gain the benefits of having a local domain controller (e.g., faster logon times and faster DNS lookup times).

This functionality is gained by the RODC introducing technologies such as the following:

- Read-only AD DS database
- Unidirectional replication
- Credential caching
- Administrator role separation
- Read-only DNS

While some of the features are not new, they may be new to the branch office that until now could not host a domain controller.

When you first launch Server Manager, you are greeted with a summary page that displays a high-level summary of the server, the roles on that server, the features that are configured on that server, and resources and support.

We will not get into every component of Server Manager, but we will highlight the Server Manager throughout the book as it relates to Active Directory. You will see that we reference it quite often as we walk through a scenario or discuss steps to perform a task.

As you will see in Figure 1.4, Server Manager contains a wealth of information about your server, specifically in the Reliability and Performance area.

Pre-Design: Microsoft Solutions Framework

When working with any design, make sure you have a good framework from which to work. Throughout the years, Microsoft has identified what it terms the Microsoft Solutions Framework (MSF), which is based on four principles:

- Work toward a shared vision.
- Stay agile; expect things to change.
- Focus on delivering business value.
- Foster open communication.

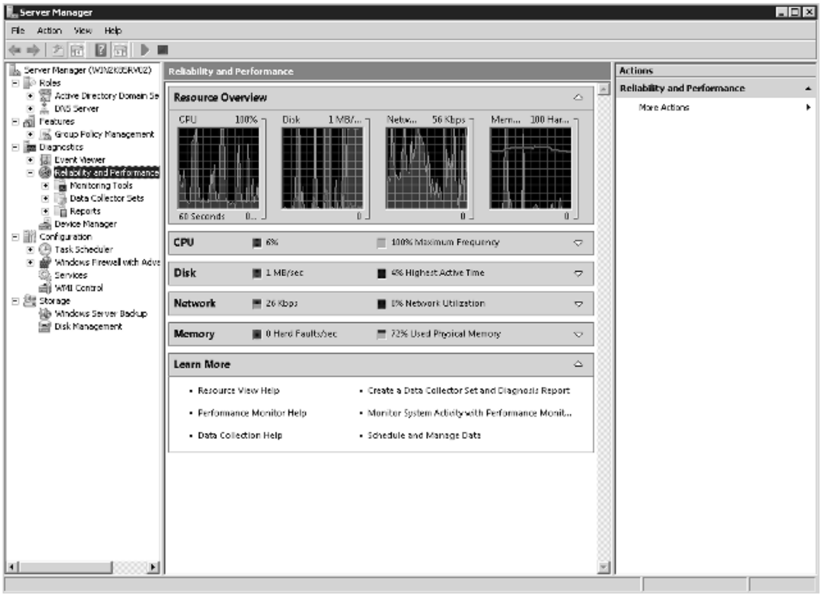


FIGURE 1.4
Server Manager

This set of guidelines can be used to control nearly any design and, if used correctly, can help stabilize the operations rollout.

When you adopt MSF, you are taking on a set of principles and models that can aid you in a successful design. When you look at the high-level view of the MSF model, you will see five distinct phases:

Envisioning

Planning

Developing

Stabilizing

Deploying

Each phase within the cycle moves one step closer to the final product, with the Envisioning phase containing the design tasks.

The Envisioning phase can then be broken down into discrete functions, the first of which should be the creation of the design team. This team will be responsible for putting together the initial design specifications and determining if the project should move forward. The roles that will be included within the design team should include individuals within six categories:

Program Management

Product Management

Development

Test

Release Management

User Experience

The Program Management role is responsible for making sure that the project is delivered on time and within budget. The team members that hold this role will need to make sure that they are on top of the overall project and are monitoring the progress. This role becomes the de facto project owner.

The Product Management role is responsible for making sure that the project meets the organization's business needs. The individuals who hold this role are responsible for making sure that the needs of the organization are met and that trade-offs in the plan are handled correctly. They will need to have a good sense of the business and understand what the customers ultimately need.

The Development role needs to have a good technical understanding of the project's design criteria and is responsible for making sure that the technical constraints of the project are met.

The Test role is responsible for making sure that the success criteria of the design is met. The Test role needs to have a good understanding of the business processes and needs to create the milestones that the design must pass to be approved.

Release Management is a role often ignored during the design phase, but it is vital to any technical rollout. The Release Management role is responsible for making sure that the piloting phase of the project moves forward without a problem. If there are problems, the Release Management role can communicate those issues with the rest of the team so that an efficient solution can be devised.

The final role is User Experience. If the users are not happy, your life will not be pleasant. The individuals that hold this role are responsible for making sure that the users' needs are met and that the design will address the need for ease of use.

While each project that goes through the design phase will have these six roles assigned to it, smaller projects may include individuals who are members of more than one team. On larger projects, you may have several members who hold a given role. No matter how many members you have for each role, make sure that the team members can perform the functions for which they are responsible, and that they know what function they are to perform. Set guidelines and, if necessary, train each member so that they have the appropriate skills.

Risk Assessment

Every project has risks involved. Risk is the possibility that you will incur some type of loss. Don't confuse the possibility of loss with the certainty of loss, however. Just because you have identified that a loss could occur doesn't mean that it *will* occur. Many projects have been stopped because nervous program managers and corporate sponsors feared that the project would cause a problem. Instead of pulling the plug on a project, the risk-assessment process should be used so that you have a basis for risk mitigation and management.

If you look at risk assessment as positive instead of negative, you can plan out the requirements to alleviate problems within the project and stop catastrophes from wiping out the budget. However, remember that risk assessment does not stop after you identify the risks. You should continually assess the risks during the entire project life cycle, because you could introduce new risk vectors as you progress.

There are six steps for you to follow during risk assessment and management:

- Risk identification—Identify the conditions that could lead to a loss and the ramifications of that loss.
- Risk analysis and prioritization—Analyze each risk and determine which risks will be considered the most dangerous or have the highest priority for the design team.
- Risk-management planning and scheduling—Develop plans that will address how risks will be controlled.

- Risk-status tracking and reporting—Continually monitor the process to identify when a risk condition has been triggered.
- Risk control—Carry out the contingency plans if a risk has been triggered.
- Risk education—Develop a database of information that will aid in the control of risks in the future.

Microsoft Solutions Framework

For more information concerning the Microsoft Solutions Framework, visit `http://www.microsoft.com/technet/solutionaccelerators/msf/default.mspx`.

Once the risk assessment is out of the way, you are ready to begin designing the Active Directory infrastructure. This can be a daunting task to undertake considering there are so many variables to consider. In the next few chapters we will introduce the knowledge necessary to build an effective design.

Coming Up Next

The next few chapters walk you through the criteria for designing a rock-solid Active Directory infrastructure. Without a good design, you will probably not have a stable infrastructure. Of course, everything within an Active Directory environment relies on a solid DNS infrastructure. If your DNS infrastructure is not stable, Active Directory will not be stable and your users will not be happy with the design.

Chapter 2, “Domain Name System Design,” outlines the requirements for a viable DNS design that Active Directory can use. The better you understand the design options, the more reliable your infrastructure will be.