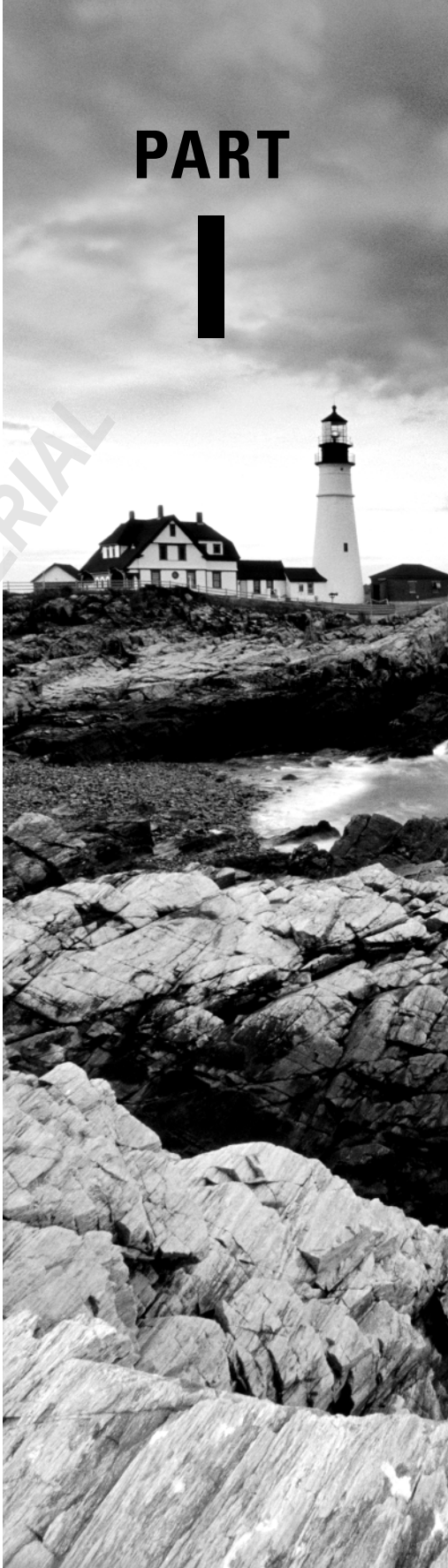
MCTS: Windows Server 2008 R2 Active Directory Configuration (Exam 70-640)

PART I



Chapter 1

Overview of Active Directory

Managing users, computers, applications, and network devices can seem like a never-ending process. As a result, you need to be organized, especially when it comes to some of the most fundamental yet tedious tasks you perform every day. That's where the concept of directory services comes in.

Microsoft's Active Directory is designed to store information about all of the objects within your network environment, including hardware, software, network devices, and users. Furthermore, it is designed to increase capabilities while it decreases administration through the use of a hierarchical structure that mirrors a business's logical organization.

You've probably also heard that a great deal of planning and training is required to properly implement Active Directory's many features. In order to reap the true benefits of this technology, you must be willing to invest the time and effort to get it right. From end users to executive management, the success of your directory services implementation will be based on input from the entire business. That's where the content of this book—and the Microsoft exams for which it will prepare you—comes in.

It's difficult to cover the various aspects of Windows Server 2008 R2's most important administrative feature—Active Directory—even in a whole book. Microsoft's main goal in Exam 70-640: Microsoft Windows Server 2008 Active Directory is to test your understanding of the features of Active Directory. The problem is that it doesn't make much sense to begin implementing Active Directory until you understand the terms, concepts, and goals behind it.

Once you have determined exactly what your Active Directory design should look like, it's time to implement it. Throughout this book, you'll learn about the various methods you can use to implement the tools and features of Windows Server 2008 R2 based on your company's business and technical requirements. Despite the underlying complexity of Active Directory and all of its features, Microsoft has gone to great lengths to ensure that implementation and management of Active Directory are intuitive and straightforward; after all, no technology is useful if no one can figure out how to use it.

In this chapter, you'll look at some of the many benefits of using directory services and, specifically, Microsoft's Active Directory. You'll explore basic information regarding the various concepts related to Microsoft's Active Directory. The emphasis will be on addressing the concepts of a directory service, why directory services are needed, the different Active Directory models, and how you can use one to improve operations in your environment. You'll then look at the various logical objects created in Active Directory and the ways in which you can configure them to work with your network environment. We will look at some of the new Windows Server 2008 R2 server roles and how they can be implemented in your company.

Finally, you'll learn the details related to how Identity and Access (IDA) in Windows Server 2008 R2 can strengthen the security of your directory services.

This book is based on Windows Server 2008 R2. Whenever we refer to Windows Server 2008 in this book, we are referring to Windows Server 2008 R2.

The Industry before Active Directory

Many production networks today are still operating without a single unified directory service. A number of small businesses and large global enterprises still store information in various disconnected systems instead of a centralized, hierarchical system such as Active Directory. For example, a company might record data about its employees (such as home addresses, phone numbers, and locations within the corporate entity) in a human resources database while network accounts reside on a Windows NT 4 Primary Domain Controller (PDC).

Other information, such as security settings for applications, resides within various other systems. And there are always the classic paper-based forms.

The main reason for this disparity was that no single flexible data storage mechanism was available. Implementing and managing many separate systems is a huge challenge for most organizations.

The Benefits of Active Directory

Most businesses have created an organizational structure in an attempt to better manage their environments and activities. For example, companies often divide themselves into departments (such as Sales, Marketing, and Engineering), and individuals fill roles within these departments (such as managers and staff). The goal is to add constructs that help coordinate the various functions required for the success of the organization as a whole.

The IT department in these companies is responsible for maintaining the security of the company's information. In modern businesses, this involves planning for, implementing, and managing various network resources. Servers, workstations, and routers are common tools of the infrastructure connecting users with the information they need to do their jobs. In all but the smallest environments, the effort required to manage these technological resources can be great.

That's where Windows Server 2008 and Microsoft's Active Directory come in. In its most basic definition, a *directory* is a repository that records and stores information and makes it available to users. Active Directory allows you to create a single centralized

(or decentralized with multiple domain controllers) repository of information with which you can securely manage a company's resources. User account management, security, and application usages are just a few of the solutions Active Directory offers. Many features of this directory services technology allow it to meet the needs of organizations of any size. Specifically, Active Directory's features include the following:

Hierarchical Organization Active Directory is based on a hierarchical layout. Through the use of various organizational components (or *objects*), a company can create a network management infrastructure and directory structure that mirror the business organization. For example, if a company called Stellacon.com had several departments (such as Sales and Human Resources), the directory services model can reflect this structure through the use of various objects within the directory (See Figure 1.1). Stellacon.com could then organize its users into the appropriate department's organizational unit (OU).

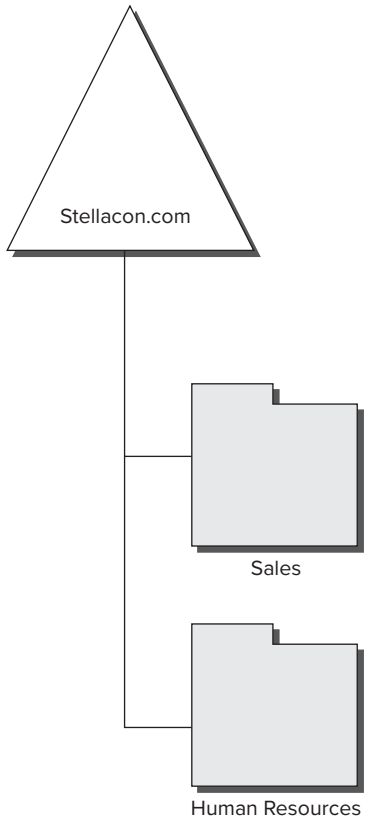
The directory structure can efficiently accommodate the physical and logical aspects of information resources, such as access to other databases, user permissions, and computers. Active Directory also integrates with the network naming service, the *Domain Name System (DNS)*. DNS provides for the hierarchical naming and location of resources throughout the company and on the public Internet.

Extensible Schema One of the foremost concerns with any type of database is the difficulty you encounter when you try to accommodate all types of information in one storage repository. That's why Active Directory has been designed with extensibility in mind. In this case, extensibility means the ability to expand (or extend) the directory schema. The *schema* is the actual structure of the database—what data types it contains and the location of their attributes. The schema is important because it allows applications to know where particular pieces of information reside. You cannot delete any portion of the schema, but you can change, modify, or alter it. The information stored within the structure of Active Directory can be expanded and customized through the use of various tools. One such tool is the Active Directory Service Interfaces (ADSI).

ADSI provides objects and interfaces that can be accessed from within common programming languages such as Visual Basic, Visual C#, and Active Server Pages (ASP). This feature allows Active Directory to adapt to special applications and to store additional information as needed. It also allows all of the various areas within an organization (or even among several organizations) to share data easily.

Centralized Data Storage All of the information within Active Directory resides within a single, distributed, data repository. Users and systems administrators must be able to easily access the information they need wherever they may be within the company. This is one of the most important design goals of the directory service—to provide a secure and centralized location for all of your data. The benefits of centralized data storage include reduced administrative requirements, less duplication, higher availability, and increased visibility and organization of data.

FIGURE 1.1 Directory services model



Replication If server performance and reliability were not concerns, it might make sense to store the entire Active Directory on a single server. In the real world, however, accessibility of remote sites and cost constraints may require that the database be replicated throughout the network. Active Directory provides for this functionality. Through the use of replication technology, Active Directory's database can be distributed among many different servers in a network environment. The ability to define sites allows systems and network administrators to limit the amount of traffic to and from remote sites while still ensuring adequate performance and usability. Reliable data synchronization allows for multimaster replication—that is, all domain controllers (except read-only domain controllers) can update information stored within Active Directory and can ensure its consistency at the same time.

Ease of Administration In order to accommodate various business models, Active Directory can be configured for centralized or decentralized administration. This gives

network and systems administrators the ability to delegate authority and responsibilities throughout the organization while still maintaining security. Furthermore, the tools and utilities used to add, remove, and modify Active Directory objects are available with all Windows Server 2008 domain controllers (except read-only domain controllers).

Network Security Through the use of a single logon and various authentication and encryption mechanisms, Active Directory can facilitate security throughout an entire enterprise. Through the process of *delegation,* higher-level security authorities can grant permissions to other administrators. For ease of administration, objects in the Active Directory tree inherit permissions from their parent objects. Application developers can take advantage of many of these features to ensure that users are identified uniquely and securely. Network administrators can create and update permissions as needed from within a single repository, thereby reducing chances of inaccurate or outdated configuration.

Client Configuration Management One of the biggest struggles for systems administrators comes with maintaining a network of heterogeneous systems and applications. A fairly simple failure—such as a hard disk crash—can cause hours of work in reconfiguring and restoring a workstation, especially an enterprise-class server. Hours of work can also be generated when users are forced to move between computers and they need to have all of their applications reinstalled and the necessary system settings updated. Many IT organizations have found that these types of operations can consume a great deal of IT staffers' time and resources. New technologies integrated with Active Directory allow for greatly enhanced control and administration of these types of network issues. The overall benefit is decreased downtime, a better end-user experience, and reduced administration.

Scalability Large organizations often have many users and large quantities of information to manage. Active Directory was designed with scalability in mind. Not only does it allow for storing millions of objects within a single domain, it also provides methods for distributing the necessary information between servers and locations. These features relieve much of the burden of designing a directory services infrastructure based on technical instead of business factors.

Search Functionality One of the most important benefits of having all your network resources stored in a single repository is that it gives you the ability to perform accurate searches. Users often see NOSs as extremely complicated because of the naming and location of resources, but they shouldn't be that complicated. For example, if we need to find a printer, we should not need to know the name of the domain or print server for that object. Using Active Directory, users can quickly find information about other users or resources, such as printers and servers, through an intuitive querying interface.

The technical chapters of this book cover the technical aspects of how Windows Server 2008 delivers all of these features. For now, keep in mind the various challenges that Active Directory was designed to address. This chapter introduces the technical concepts on which Active Directory is based. In order to better understand this topic, you'll now see the various areas that make up the logical and physical structure of Active Directory.

Understanding Active Directory's Logical Structure

Database professionals often use the term *schema* to describe the structure of data. A schema usually defines the types of information that can be stored within a certain repository and special rules on how the information is to be organized. A schema can be manipulated with the right tools, such as ADSI, mentioned earlier in the chapter. Within a *relational database* or Microsoft Excel spreadsheet, for example, we might define tables with columns and rows. Similarly, the Active Directory schema specifies the types of information that are stored within a directory.

The schema itself also describes the structure of the information stored within the Active Directory data store. The Active Directory data store, in turn, resides on one or more domain controllers that are deployed throughout the enterprise. In this section, you'll see the various concepts used to specify how Active Directory is logically organized.

Components and Mechanisms of Active Directory

In order to maintain the types of information required to support an entire organization, Active Directory must provide for many different types of functionality. Active Directory is made up of various components. Each of these components must work with the others to ensure that Active Directory remains accessible to all of the users that require it and to maintain the accuracy and consistency of its information.

In the following sections, you'll see each of the components that make up Active Directory.

Data Store

When you envision Active Directory from a physical point of view, you probably imagine a set of files stored on the hard disk that contain all of the objects within it. The term *data store* is used to refer to the actual structure that contains the information stored within Active Directory. The data store is implemented as a set of files that resides within the file system of a domain controller. This is the fundamental structure of Active Directory.

The data store itself has a structure that describes the types of information it can contain. Within the data store, data about objects is recorded and made available to users. For example, configuration information about the domain topology, including trust relationships, is contained within Active Directory. Similarly, information about users, groups, and computers that are part of the domain is also recorded.

The Active Directory data store is also commonly referred to as the Active Directory database.

Schema

The Active Directory schema consists of rules on the types of information that can be stored within the directory. The schema is made up of two types of objects: attributes and classes.

- An *attribute* is a single granular piece of information stored within Active Directory. First Name and Last Name, for example, are considered attributes, which may contain the values of Bob and Smith, respectively.
- A *class* is an object defined as a collection of attributes. For example, a class called Employee could include the First Name and Last Name attributes.

It is important to understand that classes and attributes are defined independently and that any number of classes can use the same attributes. For example, if we create an attribute called Nickname, this value could conceivably be used both as part of a User class and as part of a Computer class.

By default, Microsoft has included several schema objects. In order to support custom data, applications developers can extend the schema by creating their own classes and attributes. The entire schema is replicated to all of the domain controllers within the environment to ensure data consistency among them.

The overall result of the schema is a centralized data store that can contain information about many different types of objects—including users, groups, computers, network devices, applications, and more.

Global Catalog

The *Global Catalog* is a database that contains all of the information pertaining to objects within all domains in the Active Directory environment.

One of the potential problems with working in an environment that contains multiple domains is that users in one domain may want to find objects stored in another domain, but they may not have any additional information about those objects.

The purpose of the Global Catalog is to index information stored in Active Directory so that it can be more quickly and easily searched. The Global Catalog can be distributed to servers within the network environment. That is, network and systems administrators specify which servers within the Active Directory environment will contain copies of the Global Catalog. This decision is usually made based on technical considerations (such as network links) and organizational considerations (such as the number of users at each remote site).

You can think of the Global Catalog as something like a universal phone book. Much like the local phone book you may keep in your house, the Global Catalog is quite large and bulky, but just like the phone book, it is also very useful in helping you locate information. Your goal (as a system administrator) would be to find a balance where you are maintaining enough copies in enough locations so that users can quickly and easily access it, without it taking up too much space.

This distribution of Global Catalog information allows for increased performance of company-wide resource searches and can prevent excessive traffic across network links. Because the Global Catalog includes information about objects stored in all domains

within the Active Directory environment, its management and location should be an important concern for network and systems administrators.

Searching Mechanisms

The best-designed data repository in the world is useless if users can't access the information stored within it. Active Directory includes a search engine that users can query to find information about objects stored within it. For example, if a member of the Human Resources (HR) department is looking for a color printer, they can easily query Active Directory to find the one located closest. Best of all, the query tools are already built into Windows Server 2008 operating systems and are only a few mouse clicks away.

Replication

Although it is theoretically possible to create a directory service that involves only one central computer, there are several problems with this configuration. First, all of the data is stored on one machine. This server would be responsible for processing all of the logon requests and search queries associated with the objects that it contains. Although this scenario might work well for a small network, it would create a tremendous load on a single server in a very large environment. Second, clients that are located on remote networks would experience slower response times due to the pace of network traffic. If this server became unavailable (because of a failed power supply, for example), network authentication and other vital processes could not be carried out.

To prevent these problems, Active Directory has been designed with a replication engine. The purpose of *replication* is to distribute the data stored within the directory throughout the organization for increased availability, performance, and data protection. Systems administrators can tune replication based on their physical network infrastructure and other constraints.

An Overview of Active Directory Domains

As mentioned earlier, in a Windows Server 2008 Active Directory deployment, a domain is considered a logical security boundary that allows for the creation, administration, and management of related resources.

You can think of a domain as a logical division, such as a neighborhood within a city. Although each neighborhood is part of a larger group of neighborhoods (the city), it may carry on many of its functions independently of the others. For example, resources such as tennis courts and swimming pools may be made available only to members of the neighborhood, whereas resources such as electricity and water supplies would probably be shared between neighborhoods. So, think of a domain as a grouping of objects that utilizes resources exclusive to its domain, but keep in mind that those resources can also be shared between domains.

Although the names and fundamental features are the same, Active Directory domains are quite different from those in Windows NT. As we mentioned earlier, an Active Directory domain can store many more objects than a Windows NT domain. Furthermore, Active Directory domains can be combined together into trees and forests to form more complex hierarchical structures.

Before going into the details, let's discuss the concept of domains. If you think of a domain as a neighborhood, you can think of a group of similar domains (a *tree*) as a suburb and a group of disparate domains that trust each other (a *forest*) as a city. This is in contrast to Windows NT domains, which treat all domains as peers of each other (that is, they are all on the same level and cannot be organized into trees and forests).

Within most business organizations, network and systems administration duties are delegated to certain individuals and departments. For example, a company might have a centralized IT department that is responsible for all implementation, support, and maintenance of network resources throughout the organization. In another example, network support may be largely decentralized—that is, each department, business unit, or office may have its own IT support staff. Both of these models may work well for a company, but implementing such a structure through directory services requires the use of logical objects.

A domain is a collection of computers and resources that share a common security database. An Active Directory domain contains a logical partition of users, groups, and other objects within the environment. Objects within a domain share several characteristics, including the following:

Group Policy and Security Permissions Security for all of the objects within a domain can be administered based on policies. Thus, a domain administrator can make changes to any of the settings within the domain. These policies can apply to all of the users, computers, and objects within the domain. For more granular security settings, however, permissions can be granted on specific objects, thereby distributing administration responsibilities and increasing security.

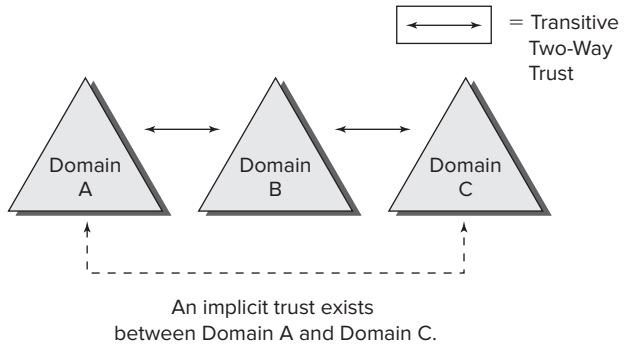
Hierarchical Object Naming All of the objects within an Active Directory container share a common namespace. When domains are combined, however, the namespace is hierarchical. For example, a user in one department might have the object name `willp@engineering.stellacon.com`, while a user in another department might have the name `wpanek@sales.stellacon.com`. The first part of the name (in these examples, the usernames `willp` and `wpanek`) is the name of the object within the domain. The suffix (in this case `engineering.stellacon.com` and `sales.stellacon.com`) is determined by the organization of the domains. The hierarchical naming system allows each object within Active Directory to have a unique name.

Hierarchical Inheritance Containers called *organizational units (OUs)* can be created within a domain. These units are used for creating a logical grouping of objects within Active Directory. The specific settings and permissions assigned to an OU can be inherited by lower-level objects.

For example, if we have an OU for the North America division within our company, we can set user permissions on this object. All of the objects within the North America object (such as the Sales, Marketing, and Engineering departments) automatically inherit these settings. The proper use of hierarchical properties allows systems administrators to avoid inconsistent security policies and makes administration easier, but it's important to remember how inheritance works when implementing and administering security, because it results in the implicit assignment of permissions.

Trust Relationships In order to facilitate the sharing of information between domains, trust relationships are automatically created between them. The administrator can break and establish trust relationships based on business requirements. A trust relationship allows two domains to share security information and objects, but it does not automatically assign permissions to these objects. *Trusts* allow users who are contained within one domain to be granted access to resources in other domains. To make administrating trust relationships easier, Microsoft has made transitive two-way trusts the default relationship between domains. As shown in Figure 1.2, if Domain A trusts Domain B and Domain B trusts Domain C, Domain A implicitly trusts Domain C.

FIGURE 1.2 Transitive two-way trust relationships



Generally, triangles are used in network diagrams to represent Active Directory domains (thereby indicating their hierarchical structure), and circles are used to represent organizational units.

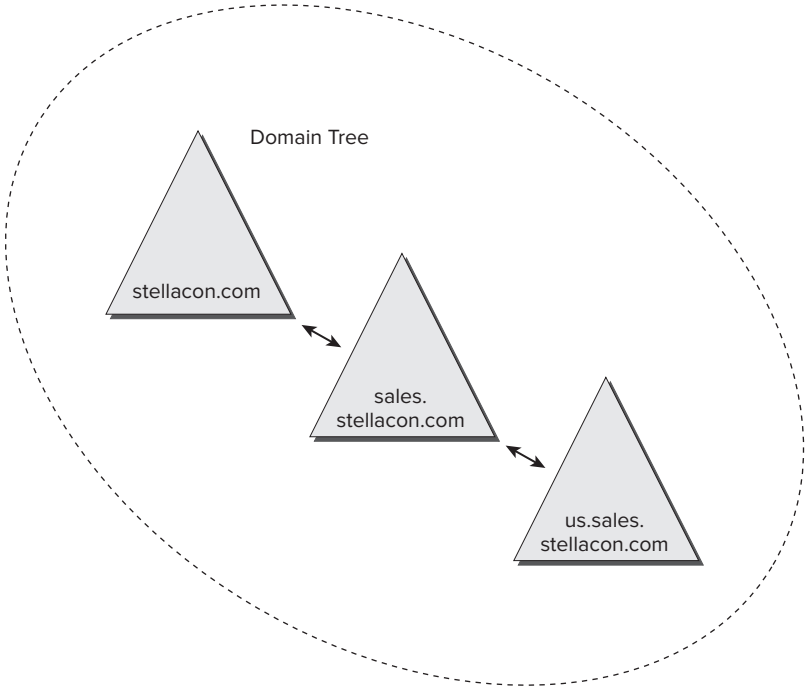
Overall, the purpose of domains is to ease administration while providing for a common security and resource database.

Overview of an Active Directory Forest

Although the flexibility and power afforded by the use of an Active Directory domain will meet the needs of many organizations, there are reasons for which companies might want to implement more than one domain. It is important to know that domains can be combined into domain trees.

Domain trees are hierarchical collections of one or more domains that are designed to meet the organizational needs of a business (see Figure 1.3).

FIGURE 1.3 A domain tree



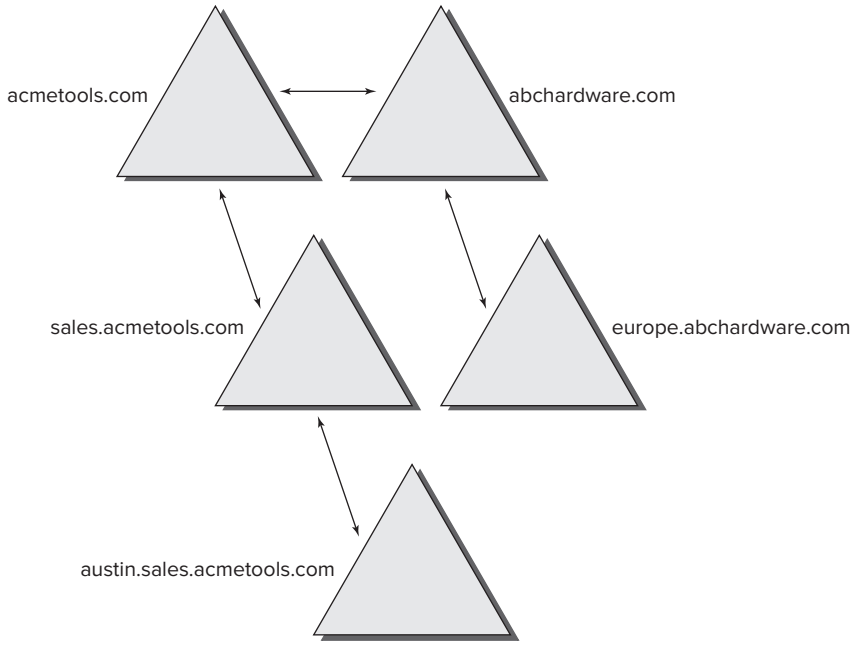
Trees are defined by the use of a contiguous namespace. For example, the following domains are all considered part of the same tree:

- `stellacon.com`
- `sales.stellacon.com`
- `research.stellacon.com`
- `us.sales.stellacon.com`

The first domain that gets installed in your Active Directory forest is called the *root domain*. Notice that all of these domains are part of the `stellacon.com` domain, which is the root domain for this tree. Domains within the same tree still maintain separate security and resource databases, but they can be administered together through the use of trust relationships. By default, trust relationships are automatically established between parent and child domains within a tree.

Although single companies will often want to configure domains to fit within a single namespace, noncontiguous namespaces may be used for several reasons. Domain trees can be combined together into noncontiguous groupings. Such a grouping is known as a *forest* (see Figure 1.4). A forest can consist of a single domain, but a forest often contains multiple noncontiguous namespaces—domains that are kept separate for technical or political reasons.

FIGURE 1.4 An Active Directory forest



Trust relationships (which facilitate shared resources) can be created among the following entities:

- Among domains within a tree
- Among trees within a forest
- Among forests (Windows Server 2003 and 2008)

Understanding Active Directory Objects

The Active Directory database is made up of units called *objects*. Each object represents a single unique database entry.

Names and Identifiers of Objects

Objects are uniquely identified within your database in the following ways:

- Each object has a globally unique identifier (GUID) or security identifier (SID).
- Each object has a distinguished name (DN).

GUIDs and SIDs

Globally unique identifiers are security identification numbers placed on applications by Active Directory. These numbers are guaranteed to be unique, but the number generated is very large, so the odds are very low that two applications will end up with the same GUID.

Security identifiers are security identification numbers placed on objects (for example, users, groups, and printers) by Active Directory. All rights and permissions are placed on the SID and not the account name. For example, let's say we have an IT manager named Maria who is going on maternity leave. John is temporarily replacing Maria. By renaming Maria's account and having John change the password, we give John all the rights and permissions that Maria had; this is because the SID on the account did not change, even though the name did.

Microsoft likes to ask questions on the exam about switching rights and permissions from one user to another. Understanding how the rights and permissions are associated with the SID will help you answer these questions correctly.

Distinguished Names

A fundamental feature of Active Directory is that each object within the directory has its own unique name, as well as a unique SID (for security objects). For example, your organization may have two different users named John Smith (who may or may not be in different departments or locations within the company). There should be some way for us to distinguish between these users (and their corresponding user objects).

Within Active Directory, each object can be uniquely identified using a long name that specifies the full path to the object. Generally, this long name for an object is called the *distinguished name (DN)*. Following is an example of a DN:

```
/O=Internet/DC=Com/DC=Stellacon/DC=Sales
/CN=Managers/CN=John Smith
```

In this name, we have specified several different types of objects:

- *Organization* (O) is the company or root-level domain. In this case, the root level is the Internet.
- *Domain component* (DC) is a portion of the hierarchical path. Domain components are used for organizing objects within the directory service. The three domain components in the example DN specify that the user object is located within the `sales.stellacon.com` domain.
- *Common name* (CN) specifies the names of objects in the directory. In this example, the user `John Smith` is contained within the `Managers` container.

Together, the components of the DN uniquely identify where the user object is stored.

Instead of specifying the full DN, you might also choose to use a *relative distinguished name (RDN)*. This name specifies only part of the object's path relative to another object. For example, if your current context is already the `Managers` group within the `sales.stellacon.com` domain, you could simply specify the user as `CN=John Smith`.

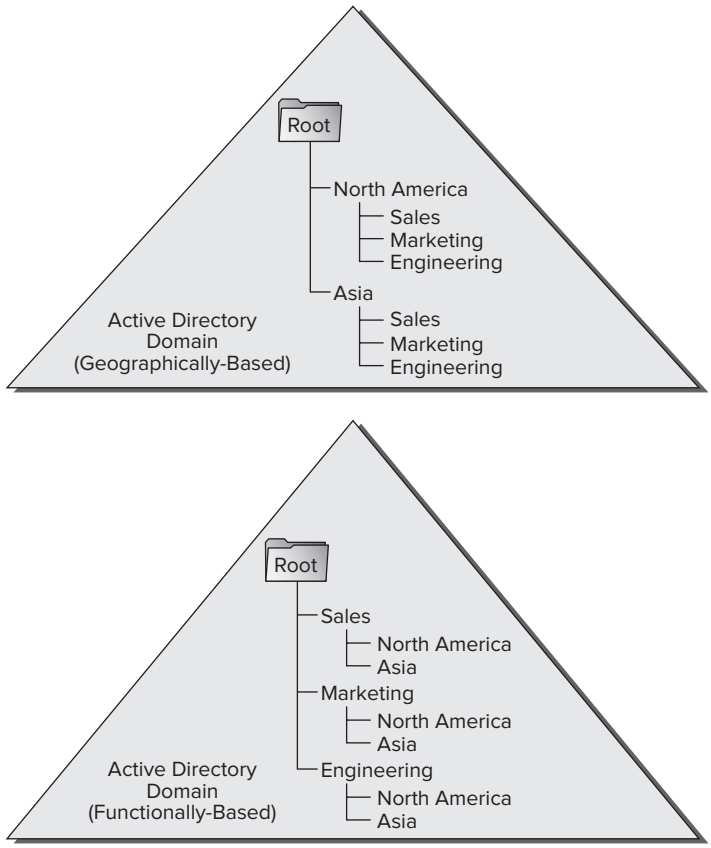
Functions of the SID and the DN

The difference between the DN and the SID is this: If you change the structure of the domain—for example, by renaming one of the containers or moving the user object—the DN of this object also changes. This type of naming system allows for flexibility and the ability to easily identify the potentially millions of objects that might exist in Active Directory.

Using Organizational Units (OUs) in Active Directory

The fundamental unit of organization within an Active Directory domain is the organizational unit (OU). OUs are container objects that can be hierarchically arranged within a domain. Figure 1.5 provides examples of two typical OU setups. OUs can contain other objects such as users, groups, computers, and even other OUs.

FIGURE 1.5 Two different OU hierarchy models



OUs are the objects to which security permissions and group policies are generally assigned. This means that proper planning of OU structure is important. A well-designed OU structure can allow for efficient administration of Active Directory objects.

OUs can be organized based on various criteria. For example, we might choose to implement an OU organization based on the geographic distribution of our company's business units or based on functional business units (see Figure 1.5).

Security Features of User, Computer, and Group Objects

The real objects that you will want to control and manage with Active Directory are the users, computers, and groups within your network environment. These are the types of objects that allow for the most granular level of control over permissions and allow you to configure your network to meet business needs.

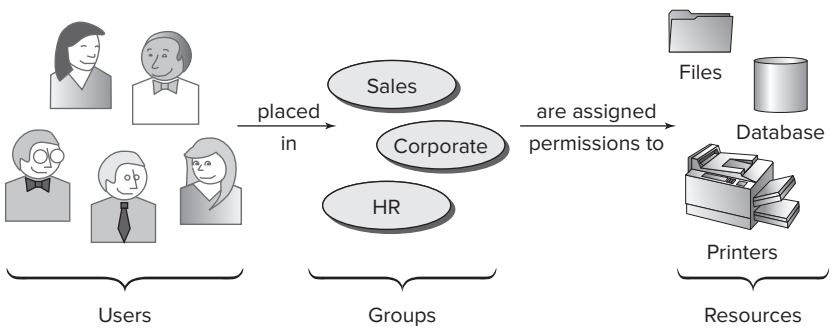
- *User account objects* enforce security within the network environment. These accounts define the login information and passwords that individuals using your network need to enter to receive permissions to use network objects.
- *Computer objects* allow systems administrators to configure the functions that can be performed on client machines throughout the environment.

Both user account objects and computer objects enable security to be maintained at a granular level.

Although security can be enforced by placing permissions directly on user and computer objects, it is much more convenient to combine users into groups for the purpose of assigning permissions.

For example, if three users will require similar permissions within the Accounting department, you can place all of them in one group and assign permissions to the group. If users are removed or added to the department, you can easily make changes to the group without having to make any further changes to security permissions. Figure 1.6 shows how groups can be used to easily administer permissions.

FIGURE 1.6 Using groups to administer security



There are two main types of groups within Active Directory:

- *Security groups* are used to administer permissions. All members of a security group receive the same security settings and are able to send email and other messages to several different users at once.
- *Distribution groups* are used only to send email and other messages to several different users at once. You don't have to maintain security permissions when using distribution groups, but they can help you handle multiple users.

Overall, using groups properly really helps you implement and manage security and permissions within Active Directory.

Delegation of Administrative Control

An OU is the smallest component within a domain to which administrative permissions and group policies can be assigned. (Administrative permissions and group policies are covered in Chapter 5, "Configuring Sites and Replication," and Chapter 6, "Configuring Active Directory Server Roles.") Now, we take a look at specifically how to set administrative control on OUs.

Delegation occurs when a higher security authority assigns permissions to a lower security authority.

As a real-world example, assume that you are the director of IT for a large organization. Instead of doing all of the work yourself, you would probably assign roles and responsibilities to other individuals.

For example, if you worked within a multidomain environment, you might make one systems administrator responsible for all operations within the Sales domain and another responsible for the Engineering domain. Similarly, you could assign the permissions for managing all printers and print queues within your organization to one individual user while allowing another individual user to manage all security permissions for users and groups.

In this way, you can distribute the various roles and responsibilities of the IT staff throughout the organization. Businesses generally have a division of labor that handles all of the tasks involved in keeping the company's networks humming. Network operating systems, however, often make it difficult to assign just the right permissions, or in other words, they have very granular permissions. Sometimes, this complexity is necessary to ensure that only the right permissions are assigned.

A good general rule of thumb is that you should provide users and administrators the minimum permissions they require to do their jobs. This way you can reduce the risk that accidental, malicious, and otherwise unwanted changes will occur.

You can also use auditing to log events to the Security Log in the Event Viewer. Doing so ensures that if accidental, malicious, and otherwise unwanted changes do occur, they are logged and traceable.

In the world of Active Directory, you use the process of delegation to define permissions for OU administrators. As a systems administrator you will occasionally need to delegate responsibility to others—you can't do it all (although sometimes some administrators believe that they can!). If you do need to delegate, remember that Windows Server 2008 was designed to offer you the ability to do so.

Simply, delegation allows a higher administrative authority to grant an individual or a group specific administrative rights for containers and subtrees. This feature eliminates the need to assign any one individual administrator sweeping authority over large segments of the user population. You can break up this control over branches within your tree, within each OU you create.

To understand delegation and rights, you should first understand the concept of access control entries (ACEs). ACEs grant specific administrative rights on objects in a container to a user or group. The container's access control list (ACL) is used to store ACEs.

When you are considering implementing delegation, there are two main concerns to keep in mind:

Parent-Child Relationships The OU hierarchy you create will be very important when you consider the maintainability of security permissions. OUs can exist in a parent-child relationship, which means that permissions and group policies set on OUs higher up in the hierarchy (parents) can interact with objects in OUs lower on the hierarchy (children). When it comes to delegating permissions, this is extremely important. You can allow child containers to automatically inherit the permissions set on parent containers. For example, if the North America division of your organization contains 12 other OUs, you could delegate the same set of permissions to all of them by placing security permissions on the North America division. By doing the task only once, you save time and reduce the likelihood of human error. This feature can greatly ease administration, especially in larger organizations, but it is also a reminder of the importance of properly planning the OU structure within a domain.

You can delegate control only at the OU level and not at the object level within the OU.

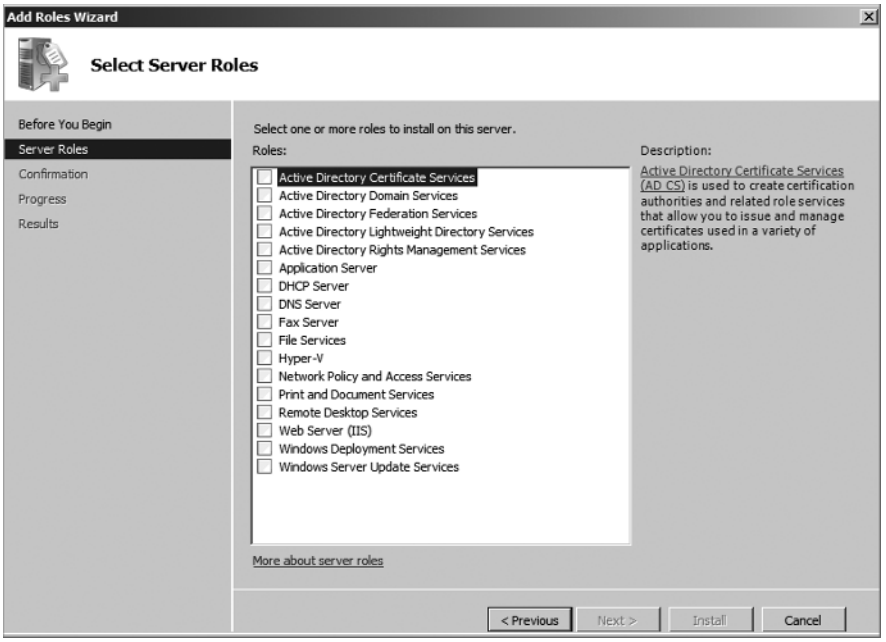
Inheritance Settings Now that you've seen how you can use parent-child relationships for administration, you should consider *inheritance*, the actual process of inheriting permissions. When you set permissions on a parent container, all of the child objects are configured to inherit the same permissions. You can override this behavior, however, if business rules do not lend themselves well to inheritance.

Introducing Windows Server 2008 R2 Server Roles

Windows Server 2003 had many tools an administrator could use to configure the services they needed to make a network run efficiently. Some of these tools included the Manage Your Server, Configure Your Server, and the Add/Remove Windows components.

Windows Server 2008 includes a feature called *Server Manager*. Server Manager is a Microsoft Management Console (MMC) snap-in that allows an administrator to view information about server configuration, the status of roles that are installed, and links for adding and removing features and roles (see Figure 1.7).

FIGURE 1.7 Windows Server 2008 R2 server roles



The following are some of the roles that you can install and manage using Server Manager.

- Active Directory Certificate Services
- Active Directory Domain Services

- Active Directory Federation Services
- Active Directory Lightweight Directory Services
- Active Directory Rights Management Services

This is not the complete list of roles. These are some of the roles that directly affect Active Directory. All Active Directory server-based roles are discussed in detail in Chapter 5.

Active Directory Certificate Services

Active Directory Certificate Services (AD CS) allows administrators to configure services for issuing and managing public key certificates. Companies can benefit from AD CS security by combining a private key with an object (such as users and computers), devices (such as routers), or services. When using Server Manager, you can configure the following components of AD CS:

Web Enrollment This feature allows users to request certificates and retrieve certificate revocation lists (CRLs) through the use of a web browser.

Certificate Authorities Enterprise Root CAs and Stand Alone Root CAs are the two types of CAs. Enterprise Root CAs (automatically integrated with Active Directory) are the top-most trusted CAs of the hierarchy. They hold the certificates that you issue to the users within your organization. The Stand Alone Root CAs hold the CAs that you issue to Internet users.

CAs below the Enterprise and Stand Alone Root CAs in the hierarchy are referred to as Subordinate CAs. The Enterprise or Stand Alone Root CAs give certificates to the Subordinate CAs, which in turn issue certificates to objects and services.

Network Device Enrollment Service The Network Device Enrollment Service allows network devices (such as routers) to obtain a certificate even though they do not have an account in the Active Directory domain.

Online Responder Service Some applications such as Secure/Multipurpose Internet Mail Extensions (S/MIME), Secure Sockets Layer (SSL), Encrypting File System (EFS), and smart cards may need to validate the status of a certificate. The Online Responder service responds to certificate status requests, evaluates the status of the certificate that was requested, and answers the request with a signed response containing the certificate's status information.

Active Directory Domain Services

In Windows Server 2008, you can use Active Directory Domain Services (AD DS) to manage objects (users, computers, printers, etc.) on a network. Many new features have

been added to AD DS that were not available in previous versions of Windows Server Active Directory. Thanks to these new features, now organizations can securely deploy and administer AD DS more efficiently. Some of the features include the following:

User Interface Improvements The updated Installation Wizard for AD DS allows it to be installed more easily. Administrators are able to locate domain controllers anywhere throughout the enterprise, and due to the improved AD DS user interface (UI), domain controllers have new options during the installation process. One of these is the ability to set up *read-only domain controllers (RODCs).*

Read-Only Domain Controllers Windows Server 2008 has a new type of domain controller called a read-only domain controller. This gives an organization the ability to install a domain controller in an area or location (on or offsite) that has limited security.

Let's imagine a hospital running Microsoft Windows Server 2003. This hospital has many affiliated physicians' offices located near it. Most likely these remote locations would not have domain controllers at their offices because administrators usually do not like to put a writable domain controller in an unsecured location. If the staff at these offices wanted to log into the hospital system, they would have to go across the WAN to be authenticated.

Now let's imagine the same hospital running Microsoft Windows Server 2008. The hospital can now place RODCs at these remote physicians' offices, which greatly improves performance for these sites.

Auditing In previous versions of Microsoft Windows Server, you had the ability to audit Active Directory by watching for successes or failures. If an individual made a successful or unsuccessful change to an Active Directory object, the attempt was logged in the Security Log. The problem with this was that, although you could view the Security Log and notice that someone accessed an object, you could not view what they might have changed in that object's attributes.

In Microsoft Windows Server 2008, you can view the new and old values of the object and its attributes.

Fine-Grained Password Policies In Microsoft Windows Server 2000 and 2003, when an organization implemented a domain-based password policy, it applied to all users in that domain. There was no inexpensive way to have individuals or groups use a different password policy. The same limitation applied to the account lockout policy. Fine-grained password policies allow an organization to have different password and account lockout policies for different sets of users in the same domain.

Restartable Active Directory Domain Services Microsoft Windows Server 2008 gives an administrator the ability to stop or restart Active Directory Domain Services. For example, administrators can do an offline defragmentation of the Active Directory database or apply security updates without needing to restart the machine. This allows other services not dependent on Active Directory (DNS, DHCP, etc.) to continue to operate while Active Directory is offline.

Database Mounting Tool In previous versions of Active Directory, if an object got deleted, an administrator had to load multiple online backups until they found the object to restore. The Windows Server 2008 Active Directory database mounting tool (`Dsamain.exe`) allows an administrator to view Active Directory data that was backed up online or through the Volume Shadow Copy Service (snapshot) at different times and then decide which online backup or snapshot to restore. This allows an administrator to quickly find and restore the data that they need.

Active Directory Federation Services

Active Directory Federation Services (AD FS) provides Internet-based clients a secure identity access solution that works on both Windows and non-Windows operating systems.

Let's imagine a user who logs into their domain when arriving at work in the morning. An authentication box asks the user for their credentials (username and password). The same user then tries to access an Internet application that operates on another network. Normally when a user from one network tries to access an application in another network, they must have a secondary username and password.

AD FS gives users the ability to do a *single sign-on (SSO)* and access applications on other networks without needing a secondary password. Organizations can set up trust relationships with other trusted organizations so a user's digital identity and access rights can be accepted without a secondary password.

Real World Scenario

AD FS in Use

Two companies have decided to work together. Company A is a retail shop that gets all of its supplies from Company B. Once these two companies decided to become partners, if they use AD FS, they can work together as if they were one company.

The companies might set up their operations so that a manager from Company A can log into an inventory database in Company B's network and order as many products as they need without approval. A lower-level employee in Company A can also log into the inventory database and place an order, but the order first has to be approved (since this level of employee does not have the rights to automatically order) by someone with the appropriate rights.

Because these companies decided to use AD FS, they can now share resources easily.

Active Directory Lightweight Directory Services

Active Directory Lightweight Directory Services (AD LDS) is a *Lightweight Directory Access Protocol (LDAP)* directory service. This type of service allows directory-enabled applications to store and retrieve data without needing the dependencies AD DS requires.

To fully understand AD LDS, you must first understand the LDAP. LDAP is an application protocol used for querying and modifying directory services.

Think of directory services as an address book. An address book is a set of names (your objects) that you organize in a logical and hierarchical manner (names organized alphabetically). Each name in the address book has an address and phone number (the attributes of your objects) associated with it. LDAP allows you to query or modify this address book.

Active Directory Rights Management Services

Active Directory Rights Management Services (AD RMS), included with Microsoft Windows Server 2008, allows administrators or users to determine what access (open, read, modify, etc.) they give to other users in an organization. This access can be used to secure email messages, internal websites, and documents.

To secure documents, Microsoft Office 2003 Professional (Word, Excel, PowerPoint, and Outlook), Microsoft Office 2007 Enterprise, Professional Plus, or Ultimate, or Microsoft Office 2010 is required.

Organizations can use AD RMS for confidential or critical information. They can design usage policy templates that can be applied directly to the confidential information.

AD RMS requires an AD RMS–enabled client. Windows Vista and Windows 7 both include the AD RMS client by default. If you are not using Windows Vista, Windows 7, or Windows Server 2008 / 2008 R2, you can download the AD RMS client for previous versions of Windows from Microsoft's Download Center.

An advantage of AD RMS is its easy installation and administration. You can install AD RMS easily through Server Manager and administer it through the MMC snap-in. AD RMS has created three new administrative roles to allow for its easy delegation throughout an organization:

- AD RMS Enterprise Administrators
- AD RMS Template Administrators
- AD RMS Auditors

Another advantage of AD RMS is its integration with AD FS, which allows two organizations to share information without needing to install AD RMS in both organizations.

We will discuss the advantages of AD RMS in Chapter 5. Chapter 5 also shows the step-by-step installation of all the server roles we have discussed in this section.

Introducing Identity and Access (IDA) in Windows Server 2008

In today's complex business world, users may have to access resources on different types of hardware, software, and devices. Because many of these systems and devices do not always communicate with each other, it is not unusual for users to have multiple identities on multiple systems.

If you have worked in the computer industry for even a short period of time, you understand that users having multiple identities and passwords for multiple systems can cause many problems. This practice can actually increase security risks due to the errors that end users can encounter by having multiple accounts.

Real World Scenario

How Users Deal with Multiple Accounts and Passwords

In today's technical world, we all have multiple usernames and passwords. I recently watched a morning news program that stated that the average person has eight sets of these. Think about it: credit card logons, online banking, websites that we visit, and many others.

Now many of us (myself included) write down all the different website usernames and passwords we use. At home, this is normally not an issue because we do not have 100 employees walking by our computer or office. The problem occurs when we use the same method at the office.

Let's say my company, Stellacon Corporation, decides to hire a good salesperson. Now when I say a good salesperson, I mean someone you can give a list of names and phone numbers to and they can make sales happen. But now they must use a computer to do their job.

This new salesperson has a username and password to log into the Microsoft Windows domain, a username and password to log into a lead-generating website, and other usernames and passwords to do their job. So what do they do? Well, if you have been in

this industry long enough, you know the stories of what users do with their credentials—they write them down and tape them to their monitor or maybe under the keyboard. My favorite story is of the person who put all their credential information on a Rolodex card and made it the first card in their Rolodex holder.

The Information Technology department needs to train their users on the importance of user credential security. If users tape their credentials to their monitor, it's our fault as IT managers; we did not train them properly. It's up to us to help make our users safe and secure.

Users' identities are an ongoing concern for most companies. This is where Identity and Access (IDA) solutions can help an organization. Through technologies and products specifically designed for IDA, organizations can manage user identities and associated access privileges. IDA solutions can be categorized into five distinct areas:

- Directory services
- Strong authentication
- Federated Identities
- Information protection
- Identity Lifecycle Management

In the following sections, we will explain these five distinct areas in more detail. Because IDA is so tightly integrated with Windows Server 2008, some of these categories were covered in the previous section, "Introducing Windows Server 2008 R2 Server Roles." Here we will explain how these previously discussed concepts interact with IDA.

Using Directory Services

As discussed earlier in this chapter, in Windows Server 2008, AD DS can be used by organizations to manage objects (users, computers, printers, etc.) on a network.

One of the advantages of using AD DS with IDA solutions is that directory services are deployed in many organizations worldwide. The chances are very good that when you work with other companies, they will also be using Microsoft directory services.

Also, by default, directory services are integrated with certificates, rights management, and Federation Services. As discussed earlier in this chapter, directory services give you the following benefits:

- Read-only domain controllers
- Auditing

- Fine-grained password policies
- Restartable Active Directory Domain Services
- Database mounting tool
- Active Directory Recycle Bin

Strong Authentication

You can strengthen your network in many ways. One of the major ways to use strong authentication is with two-factor authentication. The most common two-factor authentication method uses the *smart card*. Windows XP has built in smart-card support, but Windows Vista and Windows 7 have taken this to a higher level. Smart cards look like bank ATM cards or hotel room key cards. To use a smart card, you place it into a smart card reader and put in a personal identification number (PIN).

Another form of strong authentication uses the certificate. Certificate authority is fully integrated with Active Directory. Active Directory Certificate Services (AD CS) allows administrators to configure services for issuing and managing public key certificates. Companies can benefit from AD CS security by combining a private key with an object (such as users and computers), devices (such as routers), or services. With AD CS you get the following benefits:

- Web enrollment
- Certificate Authorities (CAs)
- Network Device Enrollment Service
- Online Responder

Strong authentication helps strengthen your IDA. Remember that IDA tries to minimize the number of usernames and passwords that users have to remember. When using a form of strong authentication, users keep track of fewer credentials (usernames and passwords) while still keeping security a top priority.

Another easy way to help with strong authentication is to enforce a strong password policy (minimum password lengths, unique characters, a combination of numbers and letters, and mixed capitalization).

Federated Identities

As we discussed earlier, AD FS gives users the ability to do a single sign-on (SSO) and access applications on other networks without a secondary password. Organizations can set up trust relationships with other trusted organizations so users' digital identity and access rights can be accepted without a secondary password.

Federated Identities enables new models for crossover SSO systems between organizations. SSO can be used for Windows and non-Windows environments.

The full implementation of Federation Identities claims–based architecture is based on the Web Services Federation (WS-Federation). The Federation Identities models support groups, roles, and rules-based models.

This works well as part of the IDA architecture because users who can use SSO authentication require fewer password resets and make fewer errors while entering credentials.

Information Protection

Active Directory Rights Management Services (AD RMS) is what information protection is all about. Information protection is included with Microsoft Windows Server 2008 R2 automatically once the AD RMS service is installed. This service allows administrators or users to determine what access (open, read, modify, etc.) they give to other users in an organization. This access can be used to secure email messages, internal websites, and documents.

Information protection supports Microsoft Office 2003 (Word, Excel, PowerPoint, and Outlook), Microsoft Office 2007, and Microsoft Office 2010.

If you are not using Microsoft Office 2003 (Word, Excel, PowerPoint, and Outlook), Microsoft Office 2007, or Microsoft Office 2010, users can always use basic information protection in the form of encryption. (Encryption is only available if the file structure is NTFS).

Information protection prevents unauthorized users from opening files, email messages, and internal websites if they do not have appropriate access. It also allows email to be tracked. This information protection and tracking will help organizations stay compliant with local, state, or federal regulations for data privacy requirements.

Some companies have designed information protection solutions to support PDF, BlackBerry, and CAD formats.

Forefront Identity Manager (FIM) 2010

The goal of Forefront Identity Manager (FIM) 2010 is to take some of the basic administration work (resetting passwords, managing groups and distribution lists, managing resource access, and policy creation) out of the hands of administrators and put it into the hands of users.

Help desk support technicians reportedly spend an average of one-third of their workdays resetting passwords.

Now I understand that many of you felt a burning in the pit of your stomach when I stated that the goal was to allow users to do administrator's tasks. For most of us, this just seems like an impossible goal. But it's not.

FIM allows you to set up policies that allow users to do specific tasks. Let's say a user goes to log into their network but they forget their password. Instead of calling IT or helpdesk, they can check a box labeled Forgot Password. A portal opens and the user is asked several security questions. If the answers are correct, the user can reset their password.

Think about a policy that allows managers to keep track of their own groups and distribution lists. Let's say a new salesperson is hired in your company. The sales manager can add that individual to the sales group and the sales distribution list. The new salesperson will now have access to all the resources every other salesperson has.

You may be thinking, "How does this help the IT department?" Well, first it helps IT save money. If administrators and IT professionals did not have to spend unproductive time doing some of these basic tasks, they could focus on important tasks.

Let's say we have an administrator who makes $70,000 a year. For a 40-hour workweek, that's about $35 an hour. Let's say that administrator spends 10 to 15 hours a week on group management and resetting passwords. That's a lot of money to pay someone to do a task that a basic user can now accomplish.

If users and managers could do some of the small day-to-day tasks that take up so much of our time, that would free us up to do some of the more important tasks we need to accomplish:

- System architecture
- System deployments
- System administration and auditing
- Creating security policies

Another advantage of FIM is that more than 20 *connectors* are included with the installation. Connectors are software add-ons that allow different applications and servers to communicate with each other. Many other add-on connectors are also available and will allow you to connect a wide range of systems and applications quickly and easily.

Summary

In this chapter, we covered Active Directory fundamentals. We gave you a high-level overview of many concepts related to Active Directory and how it is logically laid out. We covered the benefits of deploying Active Directory, including its hierarchical organization, extensible schema, centralized data storage, replication, ease of administration, network security, client configuration management, scalability and performance, and searching functionality.

We went on to cover the logical components of Active Directory, such as forests, domains, trees, and objects, and how you can create multiple Active Directory domains

and why you might do so. (For example, you can keep two companies' internal system models separate if you have a merger or acquisition.) We also covered the importance of how you name Active Directory objects and how domain naming affects the planning of Active Directory.

You then learned about the Windows Server 2008 server roles that are integrated with Active Directory. We covered the five main Windows Server 2008 Active Directory server roles (Active Directory Certificate Services, Active Directory Domain Services, Active Directory Federation Services, Active Directory Lightweight Directory Services, and Active Directory Rights Management Services).

Finally, we covered Identity and Access (IDA) solutions and how IDA can help an organization's users stay safe and secure while entering their credentials.

In the next chapter, we will cover the Domain Name System (DNS).

Exam Essentials

Understand the problems that Active Directory is designed to solve. The creation of a single, centralized directory service can make network operations and management much simpler.

Understand Active Directory design goals. Active Directory should be structured to mirror an organization's logical structure. Understand the factors that you should take into account, including business units, geographic structure, and future business requirements.

Understand Windows Server 2008 server roles. Understand what the five Active Directory Windows Server 2008 server roles—AD CS, AD DS, AD FS, AD LDS, and AD RMS—do for an organization and its users.

Understand Identity and Access (IDA) solutions. Understand how IDA can help organizations solve the problems associated with multiple usernames and passwords. Understand how the Active Directory Windows Server 2008 server roles work with and affect IDA.

Review Questions

1. Domains provide which of the following functions?
   - **A.** Creating logical boundaries
   - **B.** Easing the administration of users, groups, computers, and other objects
   - **C.** Providing a central database of network objects
   - **D.** All of the above

2. You are the administrator for a large organization with multiple remote sites. Your supervisor would like to have remote sites log in locally to their own site, but he is nervous about security. What type of server can you implement to ease their concerns?
   - **A.** Domain controller
   - **B.** Global Catalog
   - **C.** Read-only domain controller
   - **D.** Universal Group Membership Caching Server

3. Which of the following objects is used to create the logical structure within Active Directory domains?
   - **A.** Users
   - **B.** Sites
   - **C.** Organizational units (OUs)
   - **D.** Trees

4. Which of the following is *true* regarding Active Directory trust relationships?
   - **A.** Trusts are transitive.
   - **B.** By default, trusts are two-way relationships.
   - **C.** Trusts are used to allow the authentication of users between domains.
   - **D.** All of the above.

5. Which of the following protocols is used to query Active Directory information?
   - **A.** LDAP
   - **B.** NetBEUI
   - **C.** NetBIOS
   - **D.** IPX/SPX

6. You are the administrator for a large organization. Your organization currently has a Windows Server 2003 domain. Your company has set up a domain-based password policy, but the organization is unhappy with the requirement to have a single policy for all users. Your company is considering upgrading to Windows Server 2008. What feature will solve the problem of only one policy for all domain users?
   - **A.** Microsoft Windows Server 2008 multi-password policy
   - **B.** Fine-grained password policy
   - **C.** Certificate server policy
   - **D.** None of the above

7. What Windows Server 2008 server role allows a user to have a single sign-on (SSO) to access multiple applications?
   - **A.** Active Directory Domain Services
   - **B.** Active Directory Federation Services
   - **C.** Active Directory Lightweight Directory Services
   - **D.** Active Directory Rights Management Services

8. What are some of the advantages of using Windows Server 2008 Active Directory Certificate Services?
   - **A.** Web enrollment
   - **B.** Network Device Enrollment Service
   - **C.** Online Responder
   - **D.** All of the above

9. What Windows Server 2008 server role allows a user to secure an email while using Microsoft Office 2007 Outlook?
   - **A.** Active Directory Domain Services
   - **B.** Active Directory Federation Services
   - **C.** Active Directory Rights Management Services
   - **D.** Active Directory Lightweight Directory Services

10. Which of the following features of Active Directory allows information between domain controllers to remain synchronized?
    - **A.** Replication
    - **B.** The Global Catalog
    - **C.** The schema
    - **D.** None of the above

Answers to Review Questions

1. D. All of these options are features of domains and are reasons for their usefulness.
2. C. Windows Server 2008 has a type of domain controller called a read-only domain controller (RODC). This gives an organization the ability to install a domain controller in an area or location (on or offsite) where security is a concern.
3. C. OUs are used for creating a hierarchical structure within a domain. Users are objects within the directory, sites are used for physical planning, and trees are relationships between domains.
4. D. Trusts are designed for facilitating the sharing of information and have all of these features.
5. A. LDAP is the Internet Engineering Task Force (IETF) standard protocol for accessing information from directory services. It is also the standard used by Active Directory.
6. B. Fine-grained password policies allow an organization to have different password and account lockout policies for different sets of users in the same domain.
7. B. Active Directory Federation Services gives users the ability to do an SSO and access applications on other networks without a secondary password.
8. D. Web enrollment, certificate authorities (CAs), the Network Device Enrollment Service, and the Online Responder are four advantages of Active Directory Certificate Services.
9. C. Active Directory Rights Management Services (AD RMS) is included with Microsoft Windows Server 2008. This service allows administrators or users to determine what access (open, read, modify, etc.) they give to other users in an organization. This access can be used to secure email messages, internal websites, and documents. Organizations can use AD RMS for confidential or critical information.
10. A. Replication ensures that information remains synchronized between domain controllers.

Microsoft continually updates its question pool. We occasionally will add new and updated questions online. Please check the Sybex website at www.sybex.com/go/mctswindows2008r2.