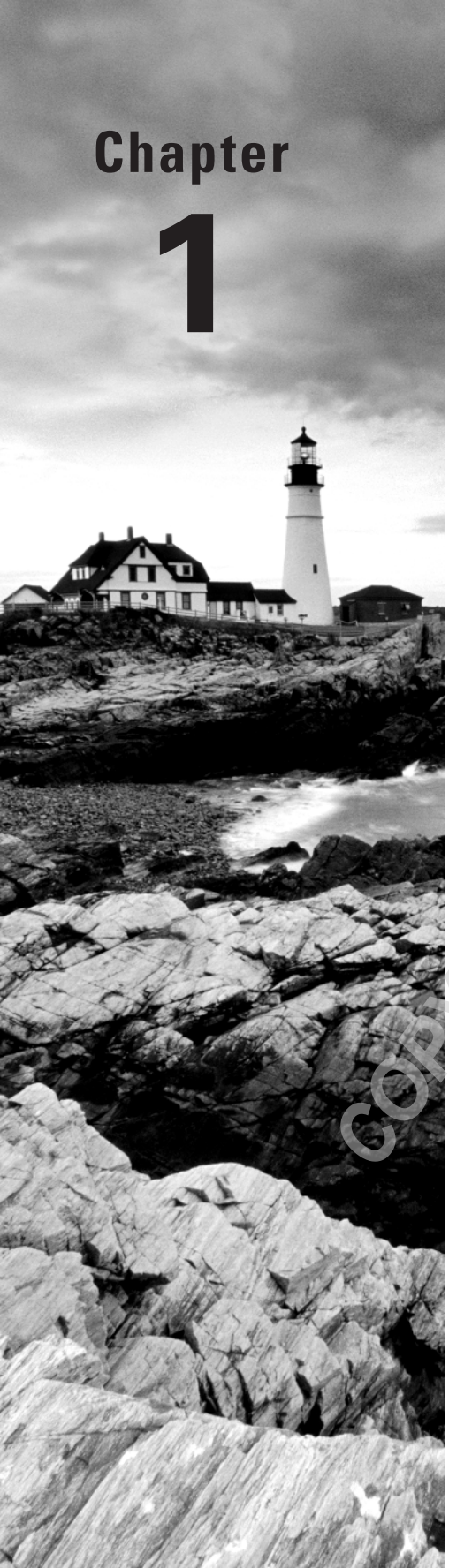


Chapter 1

Designing a Complex Windows Server 2008 Infrastructure

OBJECTIVES COVERED IN THIS CHAPTER:

✓ **Design of Active Directory forests and domains**

- May include but is not limited to: forest structure, forest and domain functional levels, intra-organizational authorization and authentication, schema modifications

✓ **Design of the Active Directory physical topology**

- May include but is not limited to: placement of servers, site and replication topology

Up until this point in your administration career, you've probably spent most of your time utilizing the technological aspects of Windows Server 2008. As you've probably read somewhere (either in another book in this series, on Microsoft's website, or in another one of your IT resources), succeeding at this exam is going to take a dramatic change from what you've gotten used to. That's because the certification exam concentrates on the idea of *design*. In other words, it's about how to create a network structure from the ground up. Thus, the beginning of this book is going to concentrate on one of the fundamental features of design: planning.

Planning is the process of realizing the needs of your network, the features that your organization will require, and the physical limitations placed on your environment, such as distance, office size, or even walls. Realistically, in the modern IT workplace, planning for an entire enterprise usually requires more than just one person. You can bet that most Fortune 500 companies make a lot of their decisions based on recommendations from a board or panel of administrators, each with their own individual experience, beliefs, and opinions. This makes a lot of sense, because when you're working with a lot of smart and experienced people on a collaborative project, you can usually create the best solution if you work together as a team, rather than doing everything by yourself.

This said, it's important for this exam (and for the real world) that you understand the overall concepts of design and most of the typical trends in the industry. To help with this, in the beginning of this chapter, I'll cover some of the basic concepts of design. I will then quickly move into what technologies are available with Windows Server 2008 and what tools you as an administrator have at your disposal. I'll end this chapter with a quick roundup of what I have covered, along with some good tips and suggestions for design.

Overview of Design Models

The first and most basic task in all designs is to determine a structure called an *administrative model*. Administrative models are conceptual logical topologies that mirror an

organization's IT administration structure. In general, one of three administrative models is usually deployed:

- Centralized
- Decentralized
- Hybrid

Please note this chapter includes some content that will not be on your exam—specifically, the discussion of the administrative hierarchy. This content, which is from MCSE exam 70-297, is covered in much more detail in Sybex's *MCSE Windows Server 2003 Active Directory and Network Structure Infrastructure Design Study Guide*. However, I'll cover each of these models and their corresponding roles in an organization. Doing this will not only help you understand the concepts of design but will also lay the groundwork for later chapters that discuss much more complicated structures.

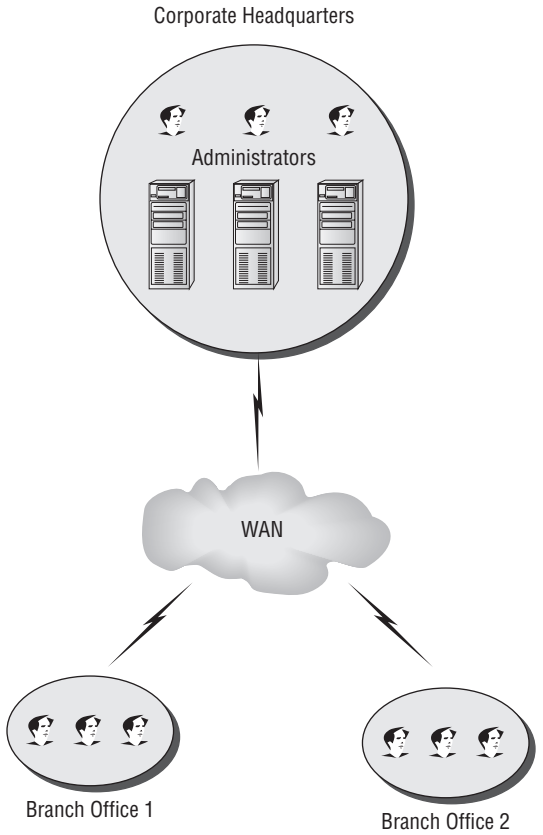
The Centralized Model

When you think of a centralized model, it's best to keep in mind the word root, *centra* (meaning center). In a pure *centralized model*, all your resources are in the same place, and all your administrators are in one spot. A good example of this would be a medium-sized business that has about 1,000 employees who each access applications, shared folders, and printers on your Microsoft network. In a centralized model, the business theoretically could exist in a large building with 20 floors, and the IT staff would all be stationed on the top floor where they could oversee all the dark deeds they must partake in on a daily basis. I'm kidding on that last part, but the concept of a pure centralized model is pretty simple. Everybody or anything who has to do with IT is in one spot!

In practice, a centralized administration model is a convenient way to run your IT function. If everything is in the same place, it's easy to get to what or whom you need to be in contact with very quickly. To top things off, you can usually get away with a lot fewer quality assurance practices because you don't have to connect across a slow WAN link to servers in your Japan office halfway around the world from your San Francisco location.

The only downside to this model is that sometimes when an organization grows, being centralized really isn't practical. Businesses open separate offices, they have remote employees, and they sometimes require IT resources in locations that don't make this realistic for a very large corporation with multiple offices. Thus, you'll fairly rarely see a pure centralized model in the wild. However, it's possible that you will see a variant of this practice called the *centralized-decentralized model*, where all of the IT *staff* is in one place but the servers are in various locations throughout the world. Figure 1.1 may help illustrate this.

FIGURE 1.1 Centralized model



The Decentralized Model

If you understand the concept of a centralized model, I'm willing to bet that you will probably be able to guess what a decentralized model is. But, just in case you can't, a *decentralized model* is an administrative model where all your IT staff and resources are *not* in the same place.

Say, for example, you work for a company called MyCorp. If MyCorp has locations in Sydney, New York, and Paris, chances are that they probably have servers in those locations. And, if MyCorp happens to be a particularly big corporation, there's an even stronger chance that MyCorp will have *multiple* servers in each of those locations. Think about it for a minute. You have at least three locations, and at each of those locations you have multiple servers. Having multiple servers probably means multiple personnel. Thus, you have a lot of different people, managing a lot of different resources, in a lot of different places!

The main reason a company will choose to implement this method is because it's a requirement. Sometimes you end up with so many resources in a place outside your main location that you need to have individuals overseeing it. And, although it's sort of a pain to not have all your resources at your disposal in one location, this model does have its upside. For one, it's incredibly scalable. With a decentralized model, you can easily add another branch or expand one of your existing branches. This makes a lot of business professionals happy, because it means that in the long term they won't be stymied by the growth of their IT administration. Instead, as the business expands, the IT administration will be able to grow along with it.

The Hybrid Administration Model

Chances are that if you understood the previous descriptions of the centralized and decentralized administration models, you will pick up on this one. A *hybrid administration model* employs some of the concepts of a centralized approach and adds some of the elegance of the decentralized approach. Earlier, when I referenced the decentralized approach and alluded to how it involved multiple administrators, each responsible for various aspects of their particular network, I failed to ask one very important key question: who manages the overall enterprise?

The hybrid administration model seeks to remedy this by combining decentralized branch administrators with a centralized oversight staff. When you break it down to its basic parts, a hybrid model is almost the same as a decentralized model. The only real difference is that the hybrid model adds another layer of authority. Whereas in the decentralized model all branches are independent, in this model the branches are independent to the extent that they are responsible for governing themselves. However, they also are dependent because a centralized group of administrators at the home office has control over all IT resources in the entire administrative structure.

In effect, this person is what you are now studying to become! An *enterprise administrator* is someone who has authority over a great number of servers throughout the organization and normally has many administrators beneath them. Instead of concentrating on an individual server or a group of servers, you will instead concentrate on the big picture: your organization's IT health. You are the one who decides the corporate structure. You are the one who creates the Group Policy standards. And you are the one who sets the model for all other administrators to follow. It's quite an important position, and it's not one that you should take lightly. If all this authority scares you, stop now! Otherwise, keep on reading. It only gets better from here.

Designing a Forest Structure

The Active Directory *forest* is the topmost design structure that you as an administrator will normally deal with. A forest contains all the domains, trees, and objects for a particular

organization. Ordinarily, most organizations use only one forest for the entire campus. However, in some large organizations, the design requirements may require multiple forests, or *forest collaboration*. Therefore, it's imperative that you as an enterprise administrator understand how a multiple-forest environment operates, as well as the domains within that environment. For our purposes, the Active Directory forest is simply a container that shares the same schema and global catalog. However, in order to design a forest for your Active Directory environment, you need to understand forest functional levels, trusts between forests, authentication within forests, and the forest schema.

Forest Functional Levels

As you've already learned from your study of Active Directory in Windows Server 2008, Windows Server 2008 currently has three functional levels at both the domain and forest levels:

- Windows 2000 Native
- Windows Server 2003
- Windows Server 2008

In previous exams, this feature wasn't quite as important as it is now. This is because each of these three functional levels has important and unique abilities, each of which is detailed in Table 1.1.

TABLE 1.1 Functional Levels

Functional Level	Available Features
Windows 2000 Native	Default Active Directory Domain Services (AD DS) features
Windows Server 2003	Forest trusts Domain renaming Linked-value replication Read-only domain controller deployment Creating dynamic objects Deactivation and redefinition of schema attributes
Windows Server 2008	No additional features, but all subsequent domain controllers will be at Windows Server 2008 level

The most important thing to know about these functional levels is that big changes will occur when you upgrade your infrastructure to the Windows Server 2003 level. Table 1.1 illustrates the drastic changes that were implemented with the release of Windows Server 2003. Beyond these major changes, the overall functional level of the forest is relatively simple.

In general, the best practice for an overall forest design is to use the most robust (and therefore highest) functional level possible. However, more often than not, because of the

presence of certain older domain controllers or technological limitations, you will be forced to settle for less advanced deployments. Keep in mind, however, that if any domains within your forest are functioning at low levels, such as Windows 2000 Native, raising the overall functional level of the forest will raise any domains that are not already at that level to at least that level.

Forest Design Elements

When you first decide to design a forest in Active Directory, you have three important elements to consider:

- Organizational requirements
- Operational requirements
- Legal requirements

As much as you may like, you're not allowed to create a forest at whim. If you did that, almost every environment would be a single forest (unless you felt curious), and it's likely that none of these requirements would be met. To understand what you need to do to accommodate these elements, let's define them a little further:

Organizational requirements is a fancy term that means "doing what your organization says needs to be done." On a practical level, what it means to administrators like you and me is that some users in the environment may require their own individual playground (an area where they can make changes and not affect anyone else), or they may require access to more folders and shared documents (items that need to be accessed by them as well as users in the rest of the structure). When you choose your design, you have to keep in mind that situations may occur where you have to assume that there may be a lot of political reasons for separation or collaboration, and you should plan accordingly.

Operational requirements (or perhaps *restrictions* is a more apt term) usually stem from the fact that different groups within a company run different services at different times. For example, a branch of your organization may run Exchange Server 2007, while another may run a custom application that conflicts with Exchange Server's requirements by directly accessing the Active Directory structure and modifying it. Thus, these branches will have to be separated at the beginning of your design.

Legal requirements are laws and regulations that impact data access throughout the network infrastructure. Believe it or not, some businesses may be legally required to keep certain functions in a separate environment. From an IT person's standpoint, this may seem kind of odd, but it's true. Imagine what would happen, for instance, if you were working for an insurance company where the claims adjusters had direct access to the accounting applications. In practical terms, this could be a bad thing. The adjusters could make payments to claimants, get direct access to underwriting data, or figure out other ways to make a mess. But in addition to these practical business reasons, legal and regulatory rules mandate separation between the accounting, underwriting, and adjusting functions.

Autonomy vs. Isolation

When you first hear the words *autonomy* and *isolation*, they might strike you as nearly synonymous. In the non-IT world, *autonomous* describes something that is independent and functions of its own accord, and *isolation* describes something that has been set apart from the rest of any given group. However, when working with Windows Networking components and building an infrastructure, these terms have very distinct and important meanings.

Autonomy

In the world of Windows administrators, *autonomy* means that a particular resource is administered, but not completely controlled by, one group. It basically means putting things together in a spot that's independently operating. This means that if you have an autonomous group of administrators, they have the rights and authority to operate on their own group of servers, or even their own domain. However, you will most likely have another group of administrators that will have authority over this group and the ability to delegate their authority (or even supercede it). In general, administrators strive for two types of autonomy when they create an administrative design: *service autonomy* and *data autonomy*. Data autonomy is the result of creating an environment where particularly important pieces of data are placed in a location that can be overseen by administrators. Service autonomy, however, is the result of making sure part of service management is directly overseen by a particular group of administrators.

Just keep in mind that in both data autonomy and service autonomy, a single group of administrators will almost always *not* be the only ones in charge.

Isolation

There's something about the word *isolation* that seems to appeal to most administrators. That's a huge stereotype, but at least in terms of the enterprise, isolation can be a very good thing. The concept behind it is that sometimes pieces of data or services running within a forest require domains that are separated completely from others. This means that, no matter what, particular administrators are the only individuals who can administer a particular area in your environment. As with autonomy, you would likely consider isolation for two resources: services and data. With *service isolation*, you are preventing other administrators from controlling a particular resource. With *data isolation*, you are blocking other administrators from accessing important files and information.

Forest Models

When designing a forest, there are traditionally three models for breaking the forest up into understandable parts: organizational, resource based, and restricted access. Each of these structures is defined in the following sections, along with the pros and cons of the concept.

Organizational Forest Model

In the *organizational* forest model, administrators design the forest from the ground up to accommodate the needs of an organization according to its departments, locations, or other criteria that define the physical layout of the campus and the functional structure of the business model. These criteria take precedence over any customized scheme provided by the network designer. As an example, the domain model shown in Figure 1.2 was designed to represent the company in a single-forest organizational model. This model is designed logically to divide the company up: first by department, and then by location.

FIGURE 1.2 Single-forest organizational model

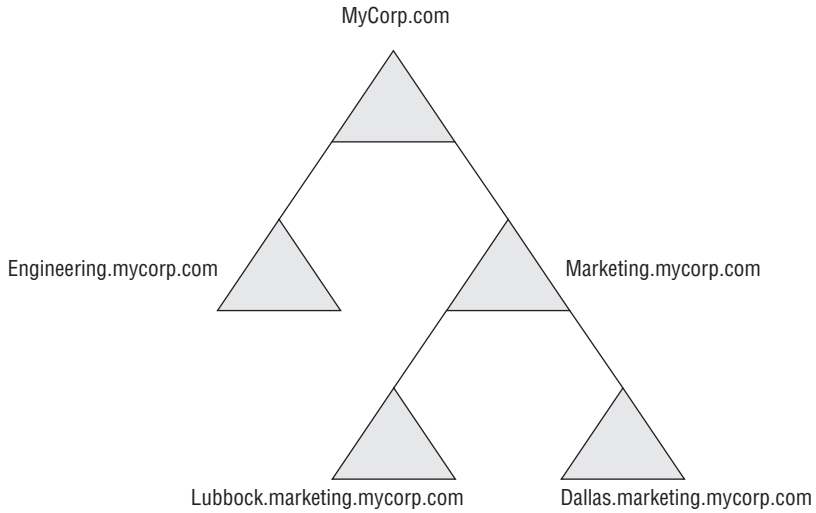
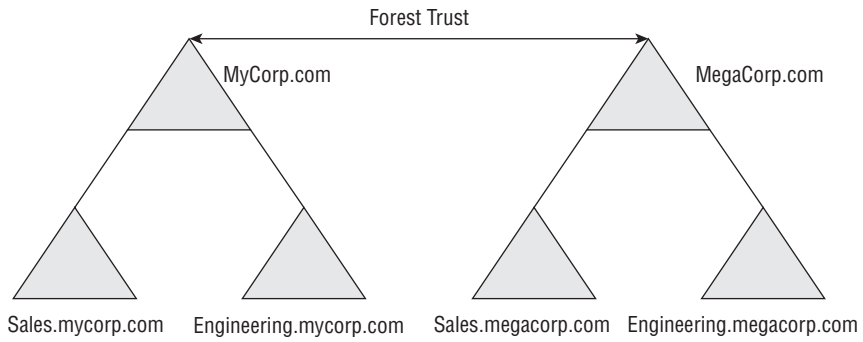


Figure 1.3 shows this same company but utilizes a trust with another forest. Each of these forests still maintains the concept of an organizational layout.

FIGURE 1.3 Organizational forest trust



Organizational forest models are useful for businesses with many different departments. Breaking up a campus by department is helpful because it alleviates the need to have a large amount of users in one location.

Resource-Based Forest Model

Fairly often companies will face situations where an environment contains valuable or highly demanding resources that have to be accessed by multiple personnel, and both users and business interests could be exposed to unnecessary liability if the resources are not separated from the rest of the organization at the forest level.

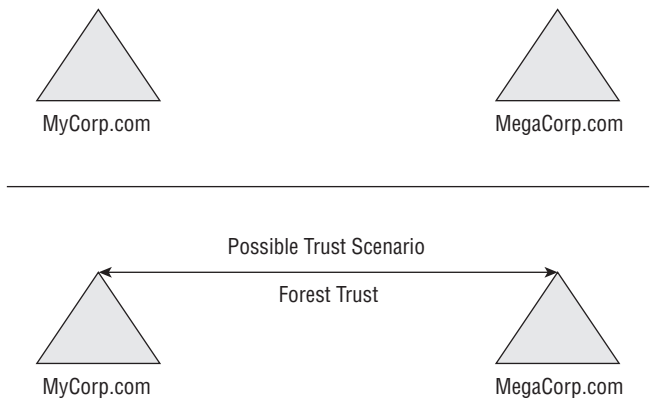
Sometimes, in an environment with a particularly useful or powerful application, shared folder, or other system resource, administrators will create a forest specifically designed for users who need to access that resource. Usually this type of forest—called a *resource-based* forest—is a new or additional forest in an organization, and trusts are established to access this forest.

Another advantage of a resource-based forest is that it is independent of any other forest; therefore, should there be a problem in a forest unrelated to the forest dedicated to the resource, the resource-based forest will be unaffected. This is particularly useful for backup strategies, which will be discussed later.

Restricted-Access Forest Model

A *restricted-access* forest, as shown in Figure 1.4, is a forest that is completely separated from another forest but (usually) is linked with a trust. This forest is administered wholly separately from the other forest and does not in any way share the administration needs with the other forest. Ordinarily, the administrators of this forest know nothing about other administration policies throughout the rest of the organization.

FIGURE 1.4 Restricted-access forest



Forest Schema

Within all versions of Windows Server, the Active Directory schema acts as a guiding rule for what exists in Active Directory, what is allowed in Active Directory, and how everything within Active Directory is formally defined. At the server level, you might see the need to modify the schema in order to incorporate different versions of Windows or new objects in Active Directory. This process can be a little annoying at times, but it's all part of upgrading your environment, which will be discussed in Chapter 2, "Naming Conventions, Networking, and Access Principles." For now, I'll cover the Active Directory schema from a higher design level and show what you need to consider before you make changes to the schema or plan for future changes.

If you have the ability to make decisions in your network, it's a good idea to create a schema policy. This determines who can upgrade the schema and when they can do it. In an environment where such a policy has not been established, administrators can upgrade and modify the schema at whim, causing potential conflicts with applications and servers, as well as a myriad of other issues.

Single-Forest vs. Multiple-Forest Design

One of the biggest advantages of working with a schema is its all-encompassing quality. Most administrators dream of the day where they can operate in a campus that uses only one forest and where they can control all the domains within. Remember, all domains with parent/child relationships that are within a forest are linked by two-way transitive trusts by default, and they're all part of the same schema. This architecture is called a *single-forest* design.

On the other hand, most organizations have a good reason to keep multiple forests, with their own independent resources, in addition to their current forest. This architecture is (surprise!) called a *multiple-forest* design.

Autonomous model An *autonomous model* is an administration concept that forces each independent forest or domain within your network to remain independent in its own administration and gives only a certain group of administrators the privilege to alter its contents. In terms of setup, this is by far the easiest model to implement. However, it can cause administrative headaches in the future because only a certain group has the authority to change permissions, and those permissions may need to be altered when the group is unavailable. Also, remember that this will affect the Active Directory application mode containers!

Collaborative Model In a *collaborative model*, administrators assist each other across multiple forests. This eases the load of administration throughout the organization. Of the two forest administration models, this is by far the more difficult one because it requires the creation of forest trusts and the delegation of privileges.

Remember, the rule of thumb is one schema per forest. In terms of the schema, a multiple-forest design will have one schema per forest in the network. This means that unless you establish a form of trust between each of your forests, they will be completely different. Thus, the only way that forests in a multiple-forest architecture can communicate is by administrator intervention. From a design standpoint, this poses a couple of questions for you to consider: Who has the right to administer the separate forests? Should the administrators in one forest have access to the neighboring forest? These questions and many others have to be answered when you are considering your forest design. One issue is whether the forests will be autonomous or collaborative.

Schema Modification

Thankfully, with all the features that have been enabled in Active Directory through the years, the process of modifying your schema has become a less common process. However, sometimes this process still does occur. In particular, it is exceptionally common when an administrator decides to install a software package that creates its own individual object classes, which may require an update to be spread throughout the rest of your Active Directory environment.

As an example, Exchange Server 2007 creates numerous individual definitions before installation that must be replicated throughout the entire environment. This creates a problem because if new object classes are being created, the schema is being modified. If you're in a large organization with a lot of users, the process of replication can take quite a while because every machine needs to become aware of what's happening throughout the rest of the environment.

Therefore, it's best to adhere to the following steps before you alter your schema:

Plan Determine what changes are required.

Plan again Make sure you've considered all the changes that are necessary.

Test your plan Simulate your changes in a test environment.

Roll out your plan Begin the changes on a small scale when traffic is low.

By following this protocol, you can make sure the process goes as smoothly as possible.

These four steps apply to almost every aspect of server design. Plan, plan again, test your plan, and roll out your plan. At the enterprise level, you can't afford to make mistakes. A single downed server can costs thousands (if not millions!) of dollars in lost productivity, transactions, or application availability. Most large organizations have specific their procedures for major upgrades and make sure the administrators follow them.

Designing an Active Directory Domain Structure

Now that I've covered most of the decisions you need to make at the forestwide level, let's take a look at what most people consider to be the more "fun" container to play with—the Active Directory domain. Domains are more fun than forests because a lot of the real administrative work goes on at this level. It's where you place most of your users, groups, and resources. It's where you assign most group policies, and most of the time it's where the servers that run all your services are. Plus, there's something really neat about browsing through Active Directory and finding the machine that's running your web box, then finding the machine that contains your DNS, and knowing full well that you can make almost anything you want happen (well, anything within the security policy, that is). You don't want to start deleting or bringing down your servers. That wouldn't be pleasant at all.

Domain Functional Levels

For each domain—just as for each forest—you have to make a decision: at which functional level does the domain need to operate? Normally, this is based on the type of servers you have and the operations that are being conducted. But remember, a domain's functional level is limited by the forest's operating level. It's easy to go up, but not easy to go down.

Just like the Active Directory forest operating level, each domain functional level has its own advantages and limitations, as shown in Table 1.2.

TABLE 1.2 Functional Level Advantages and Limitations

Domain Functional Level	Available Features	Supported Domain Controllers
Windows 2000 Native	Universal groups Group nesting Group conversion SIDs—security identifiers	Windows 2000 Server Windows Server 2003 Windows Server 2008
Windows Server 2003	`Netdom.exe` Logon timestamp updates Set `userPassword` as the effective password on `inetOrgPerson` and user objects Ability to redirect User and Computer containers Authorization Manager can store authentication policies in AD DS Constrained delegation Selective authentication	Windows Server 2003 Windows Server 2008

TABLE 1.2 Functional Level Advantages and Limitations *(continued)*

Domain Functional Level	Available Features	Supported Domain Controllers
Windows Server 2008	Distributed File System (DFS) AES encryption supported for Kerberos Last interactive logon information Fine-grained password policies	Windows Server 2008

Source: Microsoft Corporation

Single and Multiple Domains

For the purposes of this book, which is geared at the large enterprise level, it is almost not worth discussing the concept of single-domain architecture, but I will for the sake of completeness. A *single-domain* architecture is a design where within a forest there is only one single domain, usually functioning as a domain controller. At this level, most domain administrators are also enterprise administrators. It's rare to see a situation in which a large organization has only one domain with no subdomains or other administrative breakdowns.

The advantage to having a single domain is simplicity. If you have everything in one spot, it's difficult to get lost in the maze of administrative breakdowns, the schema will not be complex, and Group Policy has less of a chance (but still a significant one) of running amok.

A much more realistic design structure (one much more often seen both in the real world and on the exam) is a *multiple-domain* architecture wherein an organization has multiple websites, locations, departments, or other signifying differentiations that require the administrative structure to be broken down into simpler parts. For the remaining portion of this book, you will almost always be looking at multiple domain architectures and the roles they play in the modern workplace. See Figure 1.5 and Figure 1.6, respectively, for illustrations of a simple-domain architecture and a multiple-domain architecture.

On the certification exam, it's much more likely you will come across a more specific domain model structure, such as a regional, x, or y model.

FIGURE 1.5 Single-domain architecture

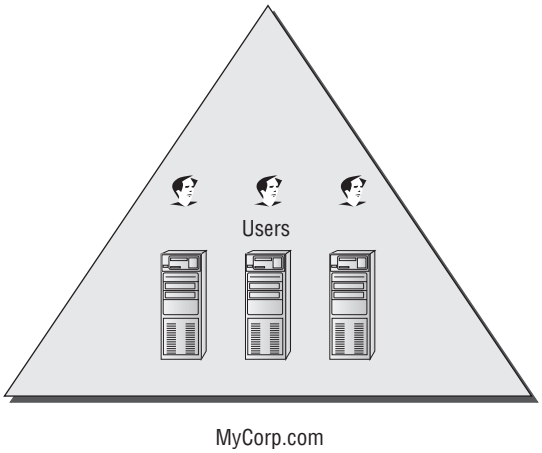
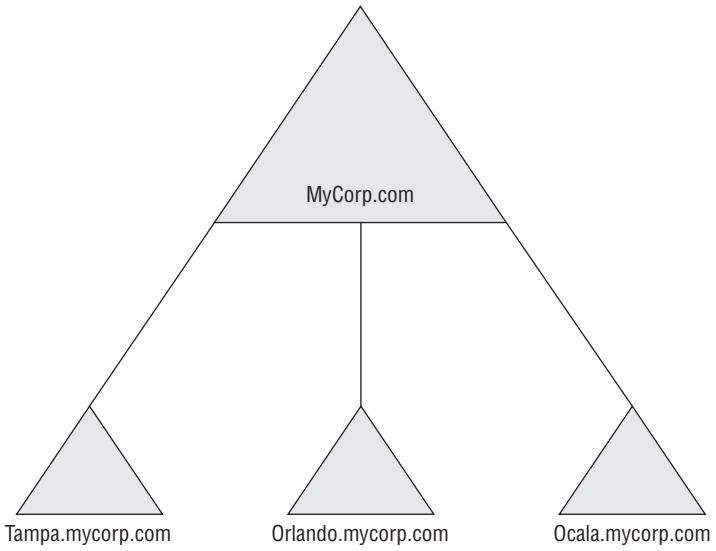


FIGURE 1.6 Multiple-domain architecture

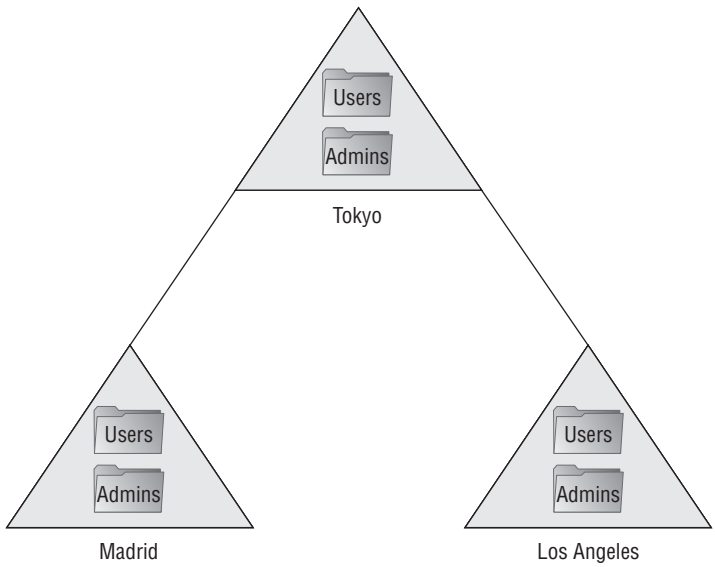


Regional Domain Model

In most international and large-scale companies, users are often divided into several geographic locations, such as Tokyo, Madrid, Hong Kong, and Los Angeles. Historically, the only way to connect these locations has been via a wide area network (WAN) connection over a relatively slow bandwidth link.

In a regional design, each of these regions is assigned their own specific domain where they can be further subdivided into more closely knit administrative groups. Figure 1.7 shows an example of this type of domain structure.

FIGURE 1.7 Regional domain model

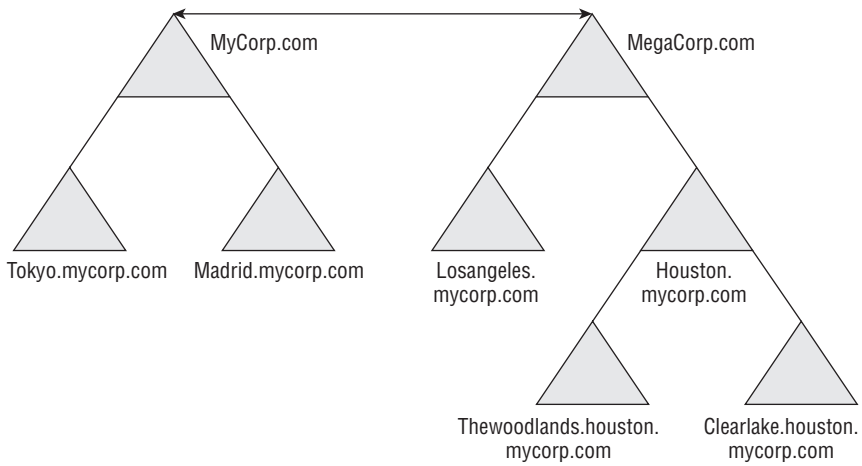


Sometimes when you need to isolate particular services using an autonomous model (not an isolation model!), it becomes necessary for you to create a multiple tree infrastructure wherein services or data are allocated among separate domain trees in a fashion that allows for a broader form of administration. You can see this model in action in Figure 1.8.

The main advantage of this model is that you manage to achieve a form of autonomous separation, but you also get to maintain the simplicity of a single schema. And if there's one aspect of Windows Server that's annoying to mess with, it's the schema.

Of course, this structure has drawbacks. Specifically, if you decide to use this form of administration, you remove the option to have complete isolation. Because the domain trees all are in the same forest, the root-level domain will have access to the rest of the trees and therefore will be able alter important information—something that you, as an enterprise administrator, may not want to have happen. Additionally, authentication paths usually take longer in this model because users have to cross separate servers to authenticate across links that are farther away.

FIGURE 1.8 Multiple-tree domain model



Creating Your Domain Structure

Now that you've seen the elements required to create an effective domain infrastructure, I'll discuss how to put them together effectively.

The process for this, once you understand the elements involved, is rather simple. Here are the steps for domain structure creation:

1. Determine the administrative model.
   - Centralized
   - Decentralized
   - Hybrid
2. Choose a domain model.
3. Choose the number of domains.
4. Assign your domain functional levels.
5. Assign a root domain.

Active Directory Authentication

One of the most important tasks when creating your overall design (if you ask your security administrator, *the* most important task) is to make sure the right people have access to the right information at the right time. In the administrative world, we call these processes *authentication* and *authorization*:

Authentication In security administration, authentication is the process of verifying a user's identity. Is John Q. Smith really John Q. Smith? Or is he another user pretending to be John Q. Smith?

Authorization Authorization is the process of determining what access a particular user has. For example, this is the process of determining whether John Q. Smith has access to the `Shared` folder on an office server located in the main building.

Overview of Forest and Domain Trust Models

No matter where you work, there will come a point in your administrative life where you simply have to break things down. As I alluded to earlier, it's rare that you will see a large enterprise using only one domain, or even one forest, to administer an entire facility. Unfortunately (or fortunately if you'd like to consider it in terms of job security), the real world is a lot more complex. Accordingly, designs and topologies become more complex as companies grow.

The main question that comes up as this process continues is this: how can you utilize resources that aren't part of your individual infrastructure? The answer, which originally came about in Windows Server 2000, is a *trust*. By now, you probably are familiar with trusts and the various types of trusts that can be implemented in Windows Server 2008. In the following sections, I will review the various types of trusts, cover their strengths and weaknesses, and discuss strategies for implementing trusts in your environment. The MCITP certification exam will ask a lot of questions on trusts from both your previous study and what you will learn here. It's a good idea to review what you've learned in the past before you take the exam. It could save your grade!

As mentioned earlier, trusts are connections—between either domains or forests—that allow various objects within Active Directory to access, modify, and utilize resources. In general, trusts exist on two levels: forest and domain.

Forest Trusts

With the release of Windows Server 2003, Microsoft made a previously unavailable function available to administrators. Forest trusts allow an administrator to connect two forests and establish a trust between them at the forest level. This is a big change from the previous iteration, which allowed this only on the domain level. Forest trusts can be either one-way, two-way, or transitive. In a two-way transitive forest, each forest trusts the other completely. Forest trusts offer several benefits, such as simplified resource access, improved authentication, improved security, and improved administrative overhead.

It's important to note that, unlike domain trusts (discussed next), forest trusts can be created only between two forests. They cannot be extended or joined to a third. This

function is slightly limiting; however, this is utilized for security purposes and for administrative reasons. By accident, an administrator could easily end up making all components of a multitiered forest trust each other completely!

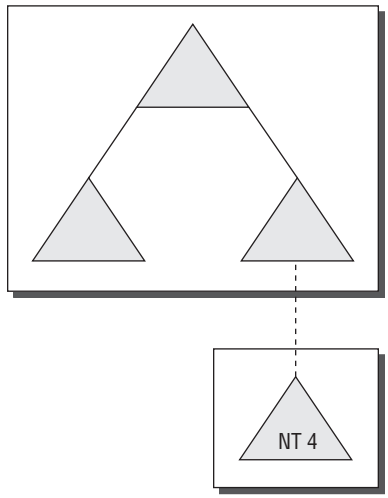
Domain Trusts

Just like at the forest level, administrators have the ability to create trusts between domains, albeit with a lot more flexibility and power than at the forest level.

In Windows Server 2008, three different trust types are available between domains in order to aid in the sharing of resources: realm trusts, external trusts, and shortcut trusts. Each of these types of trusts has various optional permissions and allowances. You will need to be familiar with them before you begin planning your infrastructure design.

External trusts You can create an *external trust* to form a one-way or two-way, nontransitive trust with domains outside of your forest. External trusts are sometimes necessary when users need access to resources located in a Windows NT 4.0 domain that doesn't support Active Directory. Figure 1.9 illustrates this.

FIGURE 1.9 External trust

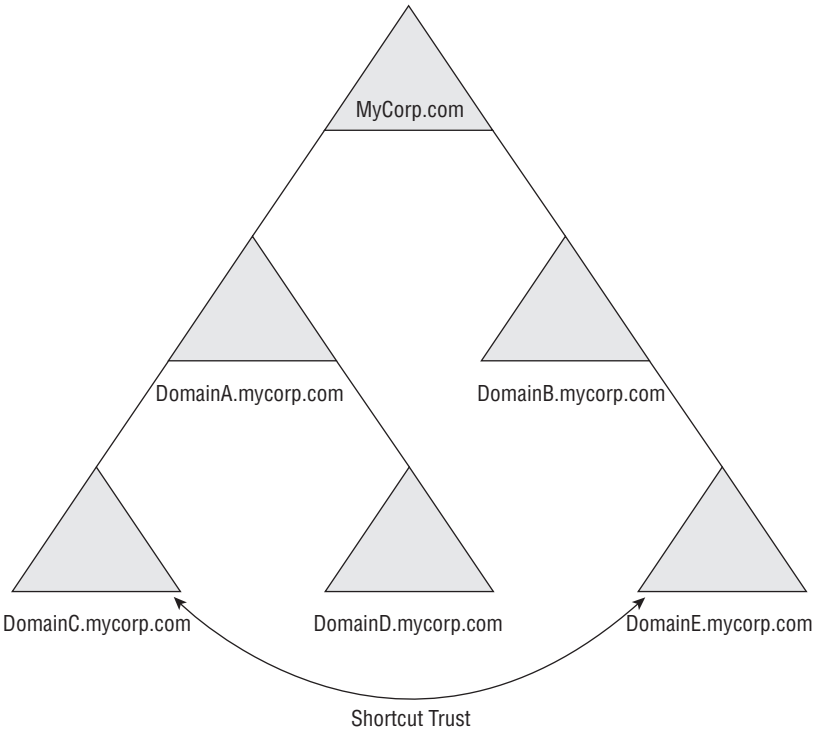


Shortcut trusts Sometimes when you have a complex Active Directory forest, the "shortest path" between two servers is not as idyllic as you might desire. If, for instance, a particular domain is nested four tiers down in your tree and it wants to access resources in another domain that is four tiers down in another tree, it will have to go up four levels and then down four levels of authentication in order to access the resources it requires.

This is quite inefficient. There is another option. By using Kerberos, you can create a transitive trust between the two domains that allows one domain to directly access another, without having to traverse up and down their various trees. This is a *shortcut trust*. It's

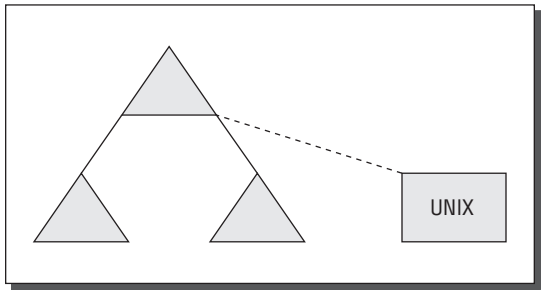
quite a useful trick, and it can save a lot of time. Keep in mind that once you create a shortcut trust, Windows Server 2008 will default to the shortest path it can to reach the desired server. This means there may come an occasion where a shortcut trust exists between a server and another server somewhere else in the network infrastructure. Undesired performance compromises can result if the server authenticates through its shortcut and then through another machine's trust. Because of this, it's best to use shortcut trusts in moderation. However, you can see a figurative example of a shortcut trust in Figure 1.10.

FIGURE 1.10 Shortcut trust



Realm trusts Since the whole world doesn't use Windows servers, it's a pretty good thing that Windows Server 2008 has a way to accommodate this. That way is a *realm trust*. Realm trusts are designed to give Unix users the ability to authenticate and have a relationship with a Windows server. This means the users on another operating system can have access to your files and resources. However, Unix realm trusts are one-way trusts and are not transitive, as illustrated in Figure 1.11.

FIGURE 1.11 Realm trust



Overview of Physical Requirements and Physical Topology

With just about everything involving computers, there are two stages: planning and implementing. So far, I have discussed some of the more basic aspects of planning, a fair share of which you may already be familiar with. The following sections will address some of the concerns you'll face with topological implementation in the modern-day workplace—issues relating to where you put the servers and how you connect them to each other.

The Microsoft MCITP enterprise administrator exam will test two things: your knowledge of both design logic and your ability to make the right decision at the right time. With regard to physical topology decisions, it usually means choosing the right type of server to be connected at the right place, running the right services and features. Accordingly, I'll start with a brief discussion of the physical characteristics of the environment that you have to consider, then discuss WAN considerations, and finally discuss specific server roles that require attention by administrators, such as the global catalog.

Restrictions

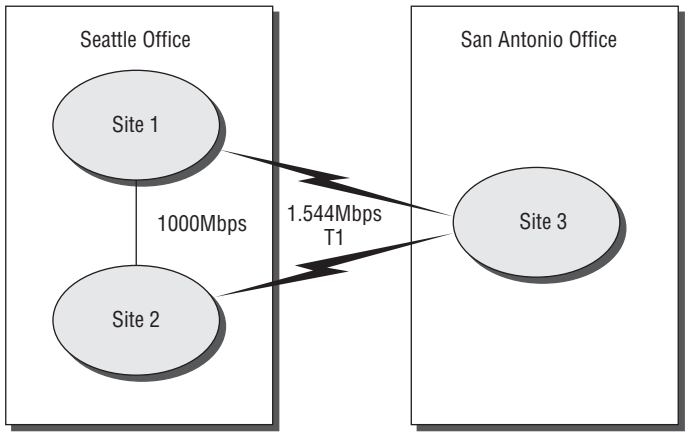
When an administrator says that they have a "restriction," it doesn't mean they're bound by the rules of emissions or dumping waste (although that beast does wander into our backyards occasionally). Instead, it usually means that something about the campus design imposes a physical limitation that hinders the speed of the data. These limitations are usually the result of geography, obstructions, or inherited design.

Geography

If parts of your environment are separated by great distances, you will most likely have a slow WAN link that connects your offices. Today this is less of a problem. As of 2008, even home users can purchase 20-megabit up and down connections for their own personal use. That

is an astonishing amount of data. However, even with the fastest WAN connections, the rule of thumb when it comes to distance still applies. If it's far away, it's probably going to be slow. Consider Figure 1.12. In this example, you have three offices; two are located in the same building in Seattle, and one is located in San Antonio. Obviously, the connection in San Antonio is going to be limited by its T1 connection.

FIGURE 1.12 Geographic concerns



Obstructions

Have you ever been asked to install a server in one location and connect to another and then been blocked by a giant wall that is 8-feet thick? If you haven't, you're the single luckiest administrator on the planet, or you simply haven't been in IT long enough to know the joys of dealing with the *other* kind of architecture. This is usually more of a building administration problem than an IT problem; however, it's good to note this, because it is a physical limitation.

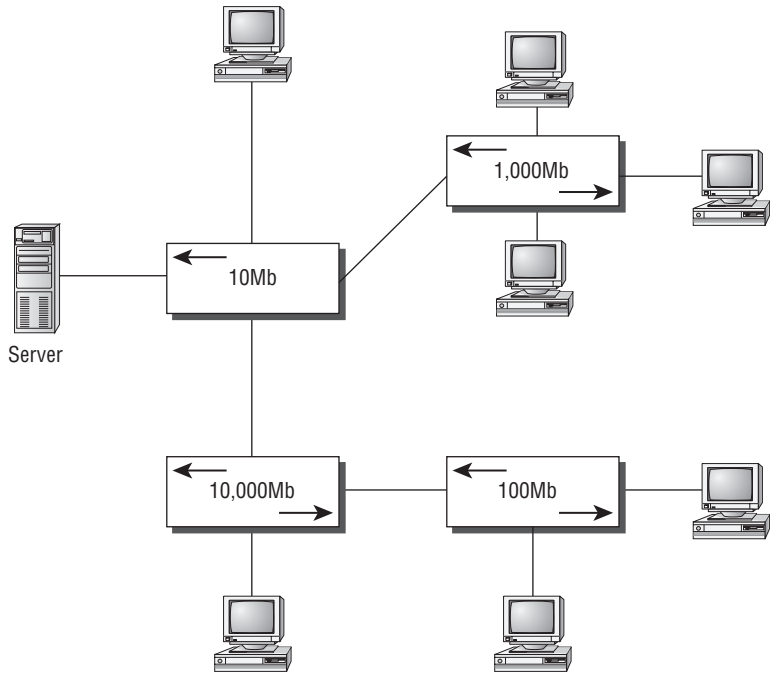
Inherited Design

This is, by far, the most frustrating of all limitations. The dreaded *inherited design* occurs when you are trying to either upgrade an environment or reassign the topology of an environment around a preexisting wiring system or topology design that has a serious bottleneck.

Here's a simple example of this: the last network engineer to work on the environment decided it would be a good idea to use only 100Mb switches instead of gigabit ones. After all, they're cheaper. But what happens when you have 100 users trying to access the same file? You're then extremely limited by how fast the server can communicate. Even though it might be capable of gigabit (or even 10Gb) performance, you're still stuck at 100Mb.

On the exam, this might sneak by you if you're not careful. Consider Figure 1.13, which shows a simple campus design. If you aren't paying close attention, it looks like a simple star topology design. However, the bottleneck of this entire environment is a relatively ancient switch. This small element can affect the entire network.

FIGURE 1.13 Inherited design concerns: the network is centered around a 10Mb switch that is slower than the rest of the network switches.



Placing Domain Controllers

The domain controller is the heart of your entire organization. It's the main location from which your users log in, and it also usually contains a copy of the global catalog, operations master, or other vital information that makes the domain controller not only the heart of your organization but several other very important organs as well.

In choosing the location for a domain controller, you should keep in mind several criteria:

Security Security is your primary concern. Since a domain controller contains so much information, you have to make sure it is not just secure regarding software but also is physically secure. Generally, this means you want to make sure the domain controller is in a safe location, such as a data center, and that a hostile intruder couldn't easily compromise your server.

Accessibility In terms of accessibility, you want to make sure this server can be accessed by your staff in terms of administration but also possesses adequate remote accessibility. Can it easily be accessed by all administrators? Are there any conflicts with the firewall or the router?

Reliability Reliability by now is probably a given. Best practices for domain controllers indicate you should always have at least one backup domain controller in an environment, and the backup controller should be ready to take over at a moment's notice. Performance, likewise, is very important. Microsoft likes to test candidates on the specific hardware requirements of Windows Server 2008 in more ways than one. In addition to requiring you

to know the basic requirements, they want you to know how much memory and disk space a domain controller will require based on the number of users. Table 1.3 breaks memory usage down into an easy-to-read format, and Table 1.4 shows how to calculate the recommended disk space, according to Microsoft.

TABLE 1.3 Domain Controller Memory Requirements

Users per Domain Controller	Memory Requirements
1–499	512MB
500–999	1GB
More than 1,000	2GB

TABLE 1.4 Domain Controller Space Requirements

Server Requirement	Space Requirement
Active Directory transaction logs	500MB
Sysvol share	500MB
Windows Server 2008 operating system	1.5 to 2GB; 2GB recommended
Intervals of 1,000 users	0.4GB per 1,000 users for the drive with `NTDS.dit`

You will have to plan for the number of domain controllers your environment will require according to the number of users, as shown in Table 1.5.

TABLE 1.5 Domain Controller Processor Requirements

Number of Users	Processor Requirements
1–499	One single processor
500–999	One dual processor
1,000–2,999	Two dual processors
3,000–10,000	Two quad processors

The Global Catalog

As you may remember from the 70-640 MCTS Windows Server 2008 Active Directory exam, a *global catalog server* is a server that contains a master list of all the objects in a domain or forest. The *global catalog* itself is the master list, and it is transmitted across servers for the purpose of informing individual machines throughout the environment of what objects actually exist and, more importantly, where they can be found. The Sybex 70-640 book *Windows Server 2008 Active Directory Configuration Study Guide* calls this list the "universal phone book" of Active Directory. Not only is that pretty clever, it's also very accurate.

The global catalog serves two more functions. First, it enables users to log on because it informs a domain controller of the universal group membership of the rest of the servers. Second, it resolves user principal names of which a particular domain controller may not be aware.

Global Catalog Server Locations

Deciding which server is going to contain your global catalog is one of the most important decisions you will make when you are beginning to design a network. Depending on its location, it can directly affect the speed of your site replication, the amount of time your servers spend updating themselves with the latest objects, and how quickly the rest of the environment becomes aware of changes.

By default, the first tree (domain) in a forest is always a global catalog server. This is because if a forest didn't have a copy of the global catalog, it really wouldn't achieve much, because no credentials would be cached and it wouldn't have a list of what user accounts existed. Beyond the initial global catalog servers, here are a couple of other reasons you might want to add a global catalog server: users of custom applications, unavailable WAN links, and roaming users.

Operations Master Location

Just like the global catalog server location, the operations master location is one of the most important design decisions you will have to make when creating your network infrastructure. However, unlike the global catalog server, the operations master server is broken down into five separate roles that have to be considered.

Schema Master

If you have a choice in the matter, the best decision for the schema master is to use it as little as possible. Modifying the schema isn't something you want to do very often, because it tends to be very heavy handed and can cause a lot of problems if you aren't careful. When placing the schema master, the main thing you have to keep in mind is the location of your schema administrators. They will be the sole benefactors of this location, and therefore you need to plan accordingly.

Domain Naming Master

In the old days of computing (the Windows 2000 era), the domain naming master was always placed on the global catalog server. With Windows Server 2008, this is no longer a requirement, but it's still considered a very good practice. The domain naming master is responsible for making sure that every domain is uniquely named and usually communicates when domains are added and then removed.

Relative Identifier Master (RID Master)

If you'll recall from your earlier study, the relative identifier (RID) and security identifier (SID) are used to distinguish uniqueness within Active Directory. Whenever an account or group of accounts is created, the RID will have to be contacted. Therefore, it's a good design practice to place the RID in an area that has access to domain controllers and can be easily communicated with whenever an administrator needs to make a new account.

Infrastructure Master

You can think of the infrastructure master in your environment as the person who makes sure everyone has everything named right. The best example of this is when someone gets married or has their name legally changed. Say a user changes her name from Maria Dammen to Maria Anderson. If she has her name changed in domain A, there is a chance that domain B may not be aware of this change and may need to be informed about it. Accordingly, something in the infrastructure has to search through everything (including network entities as well as users) and check the names and consistency.

The primary rule for placing an infrastructure master is to not place it on any machine that is a global catalog server. This is because when the global catalog server checks with other domains, it will not notice the inconsistencies between the two domains, because it's already aware of what they should be. Instead, you should try to place a domain controller in an area that is not a global catalog server but has access to a domain controller from another domain.

Primary Domain Controller

Maybe it's just a holdover from the days of old, but to this day when certain administrators hear the phrases *primary domain controller* and *backup domain controller* from their friend Windows NT 4, they shiver a little bit. This is because the world of Windows administration was nowhere near as friendly in the early days as it is now. You don't necessarily need a lecture on how things used to be, though. Instead, as a Windows Server 2008 administrator, you just need to know how to handle old equipment.

The only reason to use this role is if you have a machine running Windows NT 4 that doesn't understand Active Directory. In this situation, the primary domain controller can ensure that all older machines can change passwords as Windows Server 2008 emulates the older authentication process. Additionally, a primary domain controller makes sure that all machines with pre–Active Directory installations can keep their times synchronized.

Overview of Site and Replication Topology

In Active Directory, the concept of a site is very closely related to the concept of a subnet. A *subnet* is an isolated area in a network that is blocked by a router that stops broadcast traffic. From a design standpoint, this creates separation (and therefore isolation), and places physical firewalls between locations. The caveat to this design is that you will not have to route between IP based subnets by using a router.

Furthermore, in Active Directory the term *sites* means a collection of individual computers in a particular subnet that are logically collected into one container. This means that by default, each container will be autonomous and not communicate with any other container. To make the rest of your network communicate, you will need to establish a *site link* between the two sites within the various subnets so they can identify each other.

From a design standpoint, you are concerned with sites and subnets because of the concept of replication. As you'll recall from your study of Active Directory, *replication* is the process of notifying the rest of the network of when an object is created, deleted, moved, or changed. This is maintained by something called the *knowledge consistency checker* (KCC). The KCC generates and maintains the replication topology for replication within sites and between sites. It is a built-in process that runs on all DCs. When a system wide change takes place, the KCC (a dynamic-link library) will modify data in the local directory based on those changes and then by default, the KCC reviews and makes modifications to the Active Directory replication topology every 15 minutes to ensure propagation of such data, either directly or transitively, by creating and deleting connection objects as needed.

The KCC recognizes changes that occur in the environment and ensures that domain controllers are not orphaned in the replication topology. Due to this overhead, it is important that you take this into account when designing your site link topology and your overall infrastructure.

Site Links

Site links in Active Directory are reliable, usually WAN, connections between different subnets or collections of subnets. Remember, a site is a replication boundary. Thus, in order to communicate, you must establish a site link that connects these two different sites. Overall, each of these sites will send all their necessary replication over one individual connection, such as a T1 circuit.

Because not all site links are created equal, it behooves us as administrators to establish certain understood and quantifiable values within our site-link design:

- Site-link name
- Site-link cost
- Site-link schedule

The site-link name is pretty obvious—it's what you name your site link. The site-link cost is a little less obvious. A *site-link cost* is a value that is assigned by the administrator to identify the speed of the connection between the two different sites, with a lower number indicating a faster connection. Normally, Windows Server 2008 defaults all site links at cost 100, and it's up to administrators to manually establish costs for the rest of the topology. Table 1.6 shows a recommended cost-link table.

TABLE 1.6 Recommended Site-Link Cost Table

Available Bandwidth (Kbps)	Site-Link Cost
4096	283
2048	309
1024	340
512	378
256	425
128	486
64	567
56	586
35.4	644
19.2	798
9.6	1042

Keep in mind that site links are not limited to IP. In fact, they actually use Remote Procedure Call (RPC) over IP. But for your purposes here, IP will suffice. Site links can also use the Simple Mail Transfer Protocol (SMTP). However, SMTP is not available if you are within the same domain. Within the same domain, you are limited to RPC over IP.

The last value you need to be concerned with is the *site-link schedule*. If you read and did your exercises in Sybex's *MCTS Active Directory Configuration Study Guide*, you are probably familiar with how to set this up. Each site link requires a schedule for replication. This is because you don't necessarily want your servers replicating traffic all over the

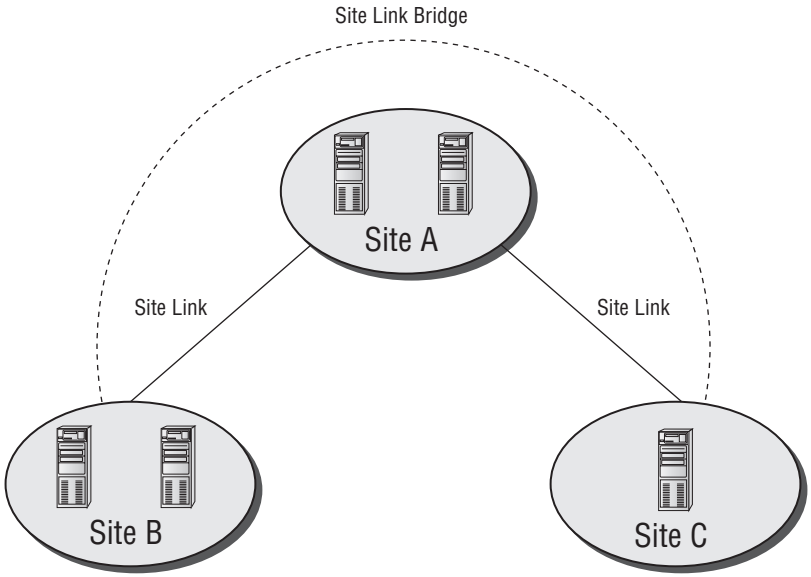
network while you have 1,000 users trying to access a particular file over a WAN. It creates a lot of traffic. For this exam, just keep in mind that schedules are a part of site links. The actual process of setting these up has already been covered.

Site-Link Bridges

The purpose of a site-link bridge is to function as a shortcut between two sites that are not actually linked together. In other words, if site A is linked to site B, and site B is linked to site C, site A can be linked through site C by using a site-link bridge.

If you have taken and passed exams 70-640 and 70-642, you are already quite familiar with this. Now you will be challenged to take what you have learned and apply it to a design environment. For instance, you may be given a scenario where you will be asked what the best solution is to connect site A to site C, as in Figure 1.14. Although it may seem like a site link would be the most logical answer, your knowledge of site-link bridges will indicate you can save some administrative overhead by using a bridge.

FIGURE 1.14 Site-link bridges



Site-link bridges can be a useful technique to implement if you already have preexisting site links in your environment. Later, in Chapter 3 of this book, I'll discuss a quick way to use this helpful feature.

Choosing Server Roles

Once you've taken the time to identify the overall structure and design scheme of your enterprise, the next logical step is to determine the role that Active Directory can play in your design. In the old days, this wasn't quite as complex a process. With systems such as Windows NT, you simply had to choose a primary domain controller (PDC), and possibly a backup domain controller (BDC), and then go through the much more arduous task of administering the environment after designing it.

Now, however, solution architects and IT administrators have the problem of not just choosing domain controllers but choosing among several types of domain controllers, DNS settings, server roles, server features, scripting, and hundreds of other options that can be implemented in your campus. Microsoft has adapted to this change in administrative design by introducing many new features and best practices that should be followed in your organization.

Part of the difficulty involved with designing an enterprise with Windows Server 2008 is that there is so much that can be done! In Windows Server 2008 alone, Microsoft released several major features that are particularly important to large environments. For the exam, you will need to be able to fire these features and their buzzwords off at whim, because you can bet a silver dollar that you're going to be seeing all them—either on the certification exam or when you enter the workforce.

Since you already have some familiarity with these new features, I'll briefly cover the most commonly tested features here and point out how you need to consider using these features in a Windows enterprise-level environment, as well as one or two old favorites that still raise their heads once in a while.

Security Design Features

Although Microsoft doesn't officially group Windows Server 2008's new features into categories such as security and delegation, it's useful for your purposes to consider the features this way, because it helps put you in the right frame of mind to think about how these features help your overall environment. The two features you'll concentrate on in the following sections are read-only domain controllers and Windows BitLocker encryption.

Read-Only Domain Controllers

As you probably already know, read-only domain controllers (RODCs) are a hot topic in the IT workplace right now. RODCs are new to Windows Server 2008 and can be run only on Windows Server 2008. However, Windows Server 2003 domain controllers can communicate with them.

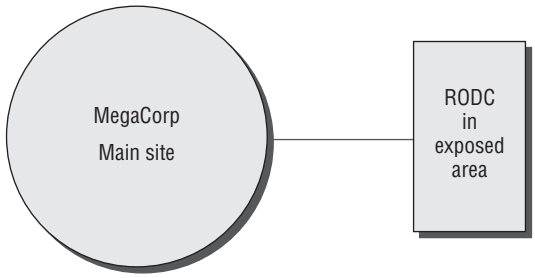
An RODC is a domain controller that, as the name implies, contains a "read-only" copy of the Active Directory database that cannot be changed (written to). The primary use for this is in a situation where the *physical* security of an environment is compromised. In other words, an RODC makes sure that someone can't steal, tamper with, or alter your domain

controller and acquire valuable intellectual property, such as usernames, passwords, or other such need-to-know information.

Something else you need to remember about RODCs is that they can use their own version of DNS, called *read-only DNS*. For design purposes, it's important to remember that read-only DNS functions by a referral system. Whenever a user makes a resolution request, the read-only DNS server makes a referral to another DNS server for name resolution. Effectively, this keeps the server from keeping a writable and updatable table of IP address mappings that can ultimately become compromised.

Figure 1.15 shows a typical example where a read-only domain controller could be used in a large enterprise environment. Keep in mind as you're studying RODCs that the RODC features fall under the purview of Active Directory Domain Services, along with the very snazzy new restartable Active Directory services feature, which I'll cover later in this book.

FIGURE 1.15 RODC placement



Windows BitLocker

Because of the release of Microsoft Vista, much of the true pizzazz of the incredible BitLocker feature has been downplayed. Why? Well, BitLocker is no longer new. However, the BitLocker feature is absolutely priceless in an enterprise environment. Single-handedly, this role ensures that your Windows server, along with its important data, is encrypted and cannot be read by a malicious intruder who nefariously acquires one of your domain controllers through deceit.

For design purposes, this feature actually opens an entire new level of consideration for the environment. Initially, you have to consider the physical placement of servers. Afterward, you have to think about whether the physical placement of those servers could possibly result in the servers being compromised. It's a rather disturbing thought, but in today's world you really just can't take the chance that your server might never be subject to malicious users.

In general, BitLocker encryption is almost never a bad thing. However, it isn't enabled by default. You, as the administrator, must turn it on. If it makes you feel safer, it isn't a bad practice to enable BitLocker encryption on almost any server that might ever have the chance of being visited by someone that isn't part of your organization. On the extreme end, you could also safely enable it on every Windows Vista and Windows Server 2008 machine in your environment. It's a bit taxing in terms of deployment but worth it in the end.

Administration and Delegation Features

It's no secret that administering several thousand users can be a big task that can take a lot of time. Not only is it a big job, but it can be rather complex. Without the ability to delegate some of the numerous tasks that have to be conducted every day, your job would be very difficult. It's because of this that nearly every day Microsoft is trying to incorporate solutions that make our lives as administrators a lot easier. Let's take a look now at some of the great new roles that you can incorporate in your design, such as Active Directory Rights Management Services (AD RMS).

Active Directory Rights Management Services

AD RMS is a true heavyweight in terms of the new features of Windows Server 2008. With AD RMS, an organization can give its users some administrative control of their individual documents and files, including office documents and emails. Additionally, with AD RMS, administrators and users can create privilege templates that can be defaulted for certain abilities in the environment. For instance, an administrator could create a "read-only" template that users could apply to a report that they'd like to make available on a shared drive somewhere in the organization.

The most important point you need to keep in mind for AD RMS in an environment is this: *Active Directory Rights Management Services requires a database server.*

If you are designing an environment and your organization doesn't plan to incorporate some sort of database server, you can't deploy this feature.

AD RMS has requirements that go beyond its hardware needs:

- Firewall exceptions (see Table 1.7)
- One AD RMS server per forest
- IIS with ASP.NET enabled

TABLE 1.7 AD RMS Port Exception Requirements

Port Exception	Description
80	HTTP port used for web traffic and communication
443	HTTPS port used for secure HTTP
1433	Microsoft SQL server port for database communication
445	SQL port for piper names

It's important to point out that AD RMS is not as simple to install as you might think. You need to keep in mind a lot of requirements, and although it might seem like a very

convenient feature for your office environment, it does require some planning on your end. You can have an AD RMS server in an extranet, in a single-server environment, connected through a URL, or with several other setups.

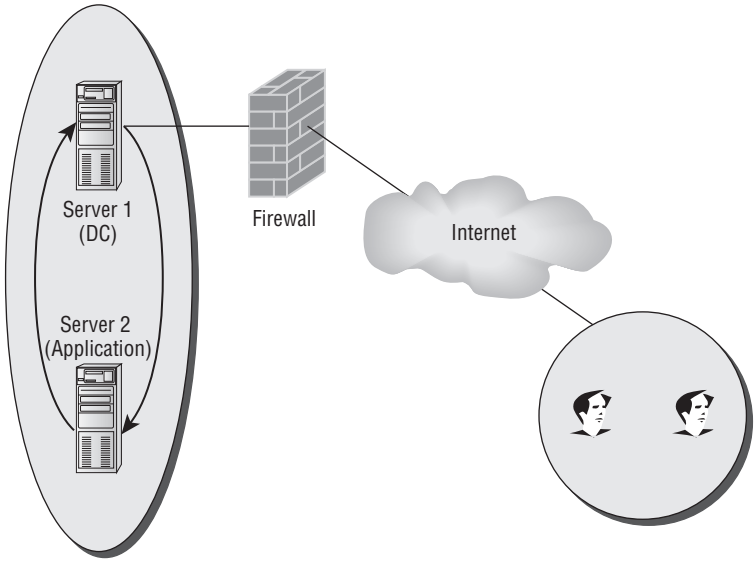
For the certification exam, it's important to remember the basic features of AD RMS—the ones in the checklist at the beginning of this section.

If you're asked for a way to give users a right to secure their documents, remember that AD RMS is probably the solution. It's extraordinarily powerful, and if you implement it correctly, it can relieve a lot of headaches. However, if you don't consider the full ramifications of this setup, you can ultimately end up with a less secure network that is running more services than it needs, ultimately draining the usability and portability of your server.

Active Directory Federation Services

If there's one phrase you need to remember about Active Directory Federation Services (AD FS), it is this: *single sign-on* (SSO). The overall design purpose of AD FS is to create an environment where users don't have to repeatedly validate their credentials across an environment. From a design standpoint, you may have a situation that looks similar to Figure 1.16.

FIGURE 1.16 Active Directory Federation Services



On the right side of the illustration, you have users who are trying to access an application on server 2. However, they authenticate through server 1. Once they've authenticated through that server, they then have to access yet another server in order to complete the tasks they need to do. As you can imagine, this can be quite annoying to users. It would be particularly irritating if they had to use the same username and password.

By using AD FS, administrators can create a trust policy between servers for the purposes of authentication. This means that in a situation such as Figure 1.16, you could create an environment where users could simply log on to their primary server and then be authenticated throughout the rest of the forest (or multiple forest) environment. It isn't just convenient for them; it's also less burdening on your servers. They get to automatically authenticate through a simple service vs. sending back and forth requests for user information that may require more demanding GUIs or other such programs they have to launch.

When you're first creating your design, Windows Active Directory Federation Services has several options on how it can be installed:

Federation Services Federation Services is the underlying architecture that provides the ability for users to sign on once in an environment. It does this through a series of designed trusts and allocations that is decided upon far in advance of the actual implementation of the feature. In general, Federation Services can implement single sign-on through one of three general federation designs, also referred to as *federation scenarios*: Web SSO, Federated Web SSO, and Federated Web SSO with Forest Trust.

Web SSO design In a simple Web SSO design, all users are external, and therefore no federation trusts exist because there are no partners. According to Microsoft, the primary reason an administrator would need a design such as this is if the organization had an application that needed to be accessed by users on the Internet.

Federated Web SSO design Sometimes companies merge, form partnerships, or otherwise need to share infrastructures and applications. Before AD FS, the only real way this could be accomplished is by creating separate accounts for each account, as well as a new series of policies and information to remember in addition to the current passwords.

Now, when situations like this occur, administrators can incorporate a design policy that implements the concept of federation trusts. A *federation trust* is a type of agreement that's made between two organizations that gives them the ability to verify users from one organization to be granted access to another. Federation trusts represented with one-way arrows point to the account side of the trust, as illustrated in 1.17.

A quick but very important point to consider before continuing is that federation trusts require two servers to authenticate. You can't have a federation trust that authenticates to nothing.

Consider the example in Figure 1.18. In this figure, you can see a great example of where an organization could use Active Directory Federation Services. MyCorp, a service providing a resource, has a trust established with MegaCorp, an organization with several accounts. Within MegaCorp, several users will need to log in to MegaCorp and have access to the services provided from MyCorp. In this scenario, they can simply log in to MegaCorp and access their applications at whim.

FIGURE 1.17 Federation trust

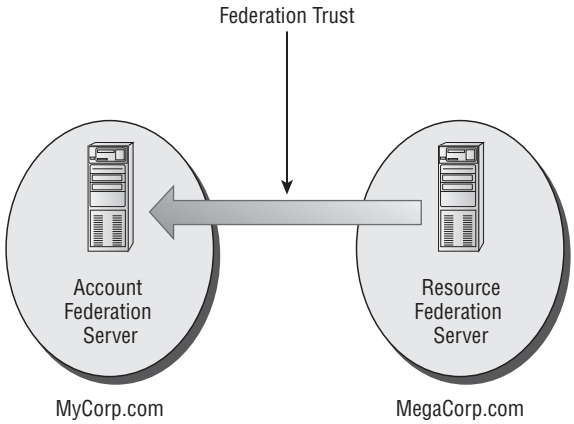
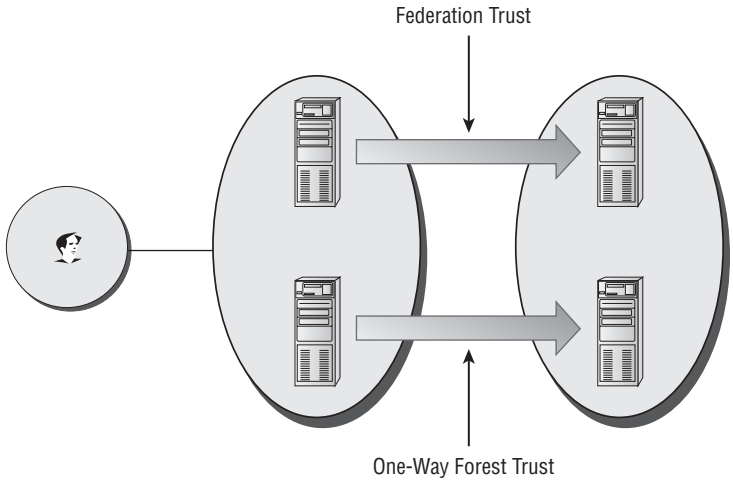


FIGURE 1.18 Federated Web SSO design



Federated Web SSO with Forest Trust design In a Federated Web SSO with Forest Trust design, you are effectively combining Active Directory from multiple forests in different organizations so that users in the account portion can access applications in another organization with their standard username and password. The advantage of this design scheme is that user accounts in the MyCorp domain can also access the application, and therefore resource accounts or groups do not need to be created.

Federation Services proxy A Federation Services proxy server is a role that serves two purposes for either the account or resource side. On the account side, a federation server acts as a proxy for the actual federation server and also distributes security tokens to web-based clients that need to access resources on the resource partner portion of the agreement. On the resource side, a proxy server just redirects clients to a federation server that can authenticate the clients. Overall, the benefit of a Federation Services proxy is to alleviate the workload of the actual federation server and add another layer of design complexity for best practices.

Claims-aware agent If an organization happens to be running an application that is claims-aware and needs to have security tokens verified through Active Directory Domains Services, they can use a claims-aware agent. Loosely put, a claims-aware application is a Microsoft ASP.NET application that uses claims that are present in an Active Directory Federation Services security token to provide authorization and personalization to your environment. Normally, you won't be implementing this unless your organization has very specific needs.

Windows token-based agent A Windows token-based agent is an agent that converts from Active Directory Federation Services to an impersonated Windows NT access token. The reason this might be used is if you have an application that requires Windows authentication and you have to connect via Federation Services. If this is the case, the web agent creates an impersonated authentication token that Windows can use. Some of the aspects of the Windows token-based agent are auditing, application logging, configuration details, and malformed requests.

If you're interested in learning more about Active Directory Federation Services and how it can be used in your environment, Microsoft provides a wonderful online resource on Windows Server 2008 TechNet that gives practical examples of how each of these designs could be implemented. However, for the purposes of the certification exam, the server-level implementation is beyond the scope of this book. What's important for you to remember is the purpose of the service and the different types of designs that can be utilized in your environment.

Summary

In this chapter, you examined the different roles of the administration process, the implementation and design of forests, and the design of the domain structure. Additionally, you examined some of the physical limitations and server roles within your infrastructure by examining WAN links, topologies, and new benefits released in Windows Server 2008.

For the exam, look over each of these individual processes, and remember the naming conventions for each theory, the features and strengths of each design structure, and what that structure will help to facilitate in your environment. In your own study, a good practice in preparing for the exam is to try to simulate a series of enterprise-level designs in your home laboratory. With just two machines, you can create a surprisingly complex amount of scenarios, such as a multiple-forest design.

Exam Essentials

Be ready to list design advantages and disadvantages. A lot of the design-based questions for the exam revolve around limitations presented in a scenario. Within the question, Microsoft will give specific hints involving the limitations or design requirements of a specific domain or forest.

Know your server 2008 features. This can't be stressed enough. Microsoft is very proud of the new features of Windows Server 2008, and truth be told, they're quite impressive. Know terms such as single sign-on, the purpose of a read-only domain controller, and where particular features can be used in your environment.

Understand the limitations of your functional levels. What is the advantage of a Windows Server 2003 domain functional level? Windows 2000 Native? In your design, you need to accommodate the different application and architecture requirements that Microsoft will require.

Know where to place your servers. If you are given a requirement for where to place your servers with a specific server role, you should be able to respond to a given diagram that shows server roles with the appropriate place in your topology to assign such a server.

Remember hardware requirements. Part of administering an enterprise is to plan for growth and understand what hardware you will need in order to run Windows for a given period of time. When planning, make sure to accommodate the eventual number of users, not just your current capacity.

Understand autonomy vs. isolation. These are not the same thing. Be able to differentiate between the two and, when asked, accommodate for the demands of each with a topology that responds well to each accordingly.

Be familiar with trust types. Know what each trust type means. Is there an advantage to a shortcut trust over another type? What is an external trust? You'll need to know the answer to these questions instantly when you take the exam.

Review Questions

1. Which of the following administrative models would be the *most* efficient for a company that maintains several corporate offices in different locations, has a single master administrator in charge of the entire campus, and has three individual administrators within each branch who are in charge of their own resources?
   - **A.** Centralized
   - **B.** Decentralized
   - **C.** Hybrid
   - **D.** Outsourced
2. Which of the following functional levels within Windows Server 2008 support read-only domain controllers? (Choose all that apply.)
   - **A.** Windows Server 2008
   - **B.** Windows 2000 Native
   - **C.** Windows Server 2003
   - **D.** Windows NT
   - **E.** Windows 2000 Mixed
3. If your design requires a single sign-on (SSO) feature to be enabled on your network, which of the following *must* be installed?
   - **A.** Active Directory Rights Management Services (AD RMS)
   - **B.** Active Directory Domain Services (AD DS)
   - **C.** Active Directory Federation Services (AD FS)
   - **D.** DNS
   - **E.** Active Directory–integrated Domain Services
4. Which of the following server feature passes the authentication of users onto a federation server?
   - **A.** Domain controller
   - **B.** Domain naming master
   - **C.** Federation proxy server
   - **D.** Relative identifier master
   - **E.** A federation trust

5. Which two of the following features need to be applied in order to make certain a domain controller in a compromised area is as secure as possible? (Choose two.)
   - **A.** Read-only domain controller
   - **B.** Active Directory Domain Services
   - **C.** Federation Services
   - **D.** Windows BitLocker
   - **E.** Security identifiers
6. Which of the following is not a component of a site link?
   - **A.** Site-link name
   - **B.** Site-link cost
   - **C.** Site-link schedule
   - **D.** Site-link identifier
7. If you needed to support Windows NT in your domain, which of the following server roles would you require?
   - **A.** Backup domain controller
   - **B.** Federation server
   - **C.** Primary domain controller
   - **D.** Domain controller
   - **E.** File server
8. Designing a domain to run completely independently of any other forest or domain structure other than itself in order to make certain it doesn't communicate with any other network is a design concept called what?
   - **A.** Centralized administration
   - **B.** Decentralized administration
   - **C.** Centralized automation
   - **D.** Autonomy
   - **E.** Isolation
9. What is the maximum number of schemas that can exist in any given forest if the administrator installs a relative identifier master on the primary domain controller?
   - **A.** 0
   - **B.** 1
   - **C.** 2
   - **D.** 3

10. Which of the following is *not* a category of requirements you should consider in designing your forest structure?
    - **A.** Organizational requirements
    - **B.** Software requirements
    - **C.** Operational requirements
    - **D.** Legal requirements

11. You are the enterprise administrator for MyCorp, a medium-sized business with 200 employees. Your superior, John Mayer, has come to you with a new design concern. In the research and development branch of the company, the engineers have been designing a new program that is designed to stress test networks and examine computers that exist throughout the infrastructure. Because of this, John has asked you to create a solution in Active Directory that will accommodate the research and development group, as well as the group's new software. Which of the following summarizes the best action to take?
    - **A.** Create a new group of users called "Research and Development." Assign a template to this group in Group Policy that restricts the usage of the network application to only those users, and then apply the policy.
    - **B.** Create a new group of users called "Research and Development." In Group Policy, require that users of the network application log onto the centralized domain controller in order to authenticate the software.
    - **C.** Create a new forest. Inside this forest, place a new group called "Research and Development" in the default domain, and adapt the isolation model.
    - **D.** Create a new domain within your forest. Inside this domain, place a new group called "Research and Development," and adapt the isolation model.

12. At a local hospital, Exchange Server 2007 is almost constantly in use. Because most medical records are secure pieces of information, these records are highly sensitive, but occasionally doctors will have to communicate with other physicians via email regarding patients under an extremely restrictive security policy, governed by legal documents. Accordingly, you have been asked to deploy a domain controller solution at most hospital branches that will allow doctors to log on at will in potentially vulnerable environments. Security is a must, and physicians must be able to access email. Which of the following is the best solution?
    - **A.** Deploy domain controllers at all branch locations with Windows BitLocker enabled. Enable single sign-on (SSO) with AD FS and require certificates to be used for each user.
    - **B.** Deploy a read-only domain controller in the areas that may be exposed and ensure that certificates are used to protect sensitive data.
    - **C.** Enable Active Directory Rights Management Services (AD RMS), and enable each physician to secure their own files. Additionally, enable Windows BitLocker.
    - **D.** Deploy a read-only domain controller, and enable Windows BitLocker.

13. Phil, a new user in your engineering department, has been tasked with creating a new piece of hardware that drills microscopic holes in pieces of fiber. According to Phil, the budget for this project is in excess of $2 million. Additionally, Phil has asked that he be allowed to administer his own individual Windows NT server so that he can accommodate legacy user demands with the device. As the lead administrator for this 12,000-person company, what design choice would be your best decision?

    A. Create a new forest. Inside this forest assign the root domain to Phil, and give Phil administrative privileges.

    B. Implement a read-only domain controller. Add Phil as the administrator of this domain controller, and add the backup domain controller feature.

    C. Create a new forest. Inside the forest, create a new domain, and make Phil the administrator of this domain.

    D. Create a new domain in your existing forest, and enable the primary domain controller emulator. Then, implement the design model for autonomy.

14. In your large enterprise, your administrators have become constantly burdened by the need to keep up with the excessive amount of file permission reassignment that is required for certain important documents, including Excel spreadsheets and email. Accordingly, you are seeking to implement an elegant solution to this problem. What would you recommend?

    A. Enable Active Directory Rights Management Services (AD RMS), and allow users to assign their own file permissions.

    B. Create a template for each file situation, and enable delegation of this template to individual users in your environment.

    C. Elevate the authority of your standard user to server operators, allowing them to assign their own policies.

    D. Enable Active Directory Domain Services with DNS enabled.

15. Within your organization, you have three sites: Tokyo, Madrid, and New York City. From Tokyo to New York, you have a site link running over a T3 line at 45Mbps. From New York to Madrid, users are connected via a 1.544Mb T1 line. To connect Tokyo to Madrid, what should you recommend to your network administrator?

    A. Create a new site link between Tokyo and Madrid.

    B. Create a site-link bridge by maintaining a transitive link between the two existing site links.

    C. Enable remote logins for your Tokyo users, and extend a two-way transitive trust from your Tokyo to Madrid location.

    D. Enable remote logins for your Madrid users, and extend a two-way transitive trust from your Tokyo to Madrid location.

16. MyCorp is a larger enterprise supporting more than 10,000 users. Recently, the CEO of MyCorp has created another business within MyCorp called MyLittleCorp. As the enterprise administrator, your CEO has asked you to install a new domain controller for your organization that will support 400 users. Which of the following hardware requirements should you make sure to maintain?

A. At least two processors

B. More than 1GB of RAM

C. At least 512MB of RAM

D. RAID 5

17. If a forest is differentiated by the Engineering, Accounting, Human Resources, and Products groups, what administrative structure is it following?

A. Organizational

B. Resource

C. Restricted access

D. Single sign-on (SSO)

18. If a forest is differentiated into the Engineering, Accounting, Human Resources, and Products groups, what administrative structure is it following?

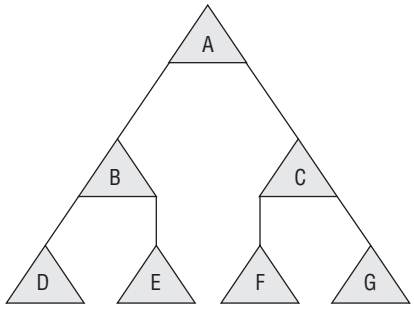
A. Organizational

B. Resource

C. Restricted access

D. Single sign-on (SSO)

19. Look at the illustration shown here. If domain D needed to communicate with domain G, which of the following trusts would you recommend?



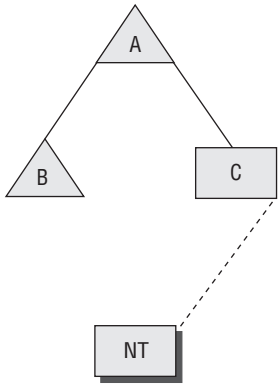
A. External trust

B. Shortcut trust

C. Realm trust

D. Friend trust

20. Consider the illustration shown here. If domain A needed to communicate with domain C, which of the following trusts would be best recommended?



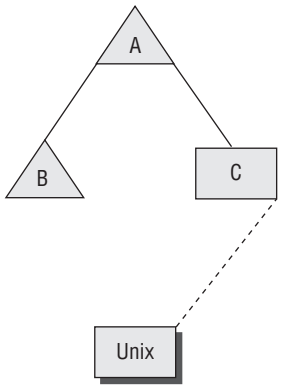
A. External trust

B. Shortcut trust

C. Realm trust

D. Friend trust

21. Consider the illustration shown here. If domain A needed to communicate with domain C, which of the following trusts would be best recommended?



A. External trust

B. Shortcut trust

C. Realm trust

D. Friend trust

22. Which of the following technologies is responsible for verifying a user's identity across a network infrastructure, and what is the name of this process? (Choose all that apply.)

A. Domain controller

B. Domain name server

C. Realm authenticator

D. Operations master

E. Authentication

F. Authorization

Answers to Review Questions

1. C. Hybrid administrative models are used in environments where there needs to be a centralized corporate administrator (or administrators), as well as several decentralized locations where other administrators can maintain their own branches.

2. A, C. Windows Server 2008 supports three functional levels: Windows Server 2008, Windows Server 2003, and Windows 2000 Native. Windows NT and Windows 2000 Mixed do not exist. Of those three functional levels, only Windows Server 2008 and Windows Server 2003 support read-only domain controllers.

3. C. Active Directory Federation Services (AD FS) is a service that allows multiple users to authenticate once in a Windows environment and then access resources on the rest of the environment. Most often, it is used for Internet users.

4. C. Federation proxy servers authenticate users and pass on their credentials to Active Directory federation servers in order to reduce server load and increase security as the proxy services provide a security token for the authentication within the corporate network.

5. A, B. By using read-only domain controllers, the domain controller in the exposed area will not be able to be written to and will cache credentials. Additionally, if that server is physically compromised, Windows BitLocker will keep the server from having its hard disk exposed to data extraction.

6. D. The three main components of a site link are the site-link name, cost, and schedule. The name identifies the site link, the cost indicates the speed and priority of the connection, and the schedule indicates when the site can be used to replicate across the network. At the enterprise level, you have to make sure site links are assigned the appropriate speed so that they are as efficient as possible.

7. C. A primary domain controller emulator is used to communicate with pre–Active Directory systems in order to support the legacy software with Windows Server 2008 technology.

8. E. The process of isolation is ensuring that an individual piece of the network can be accessed by nothing else in the entire network. This is usually done to isolate pieces of software or volatile user accounts from the rest of the infrastructure.

9. B. Remember, the rule of thumb with schemas is that there is only one schema per forest. If there were multiple schemas per forest, it would be really bad because there would be constant conflicts over the overall design of the infrastructure.

10. B. Software requirements are definitely part of the design consideration process, but they are covered under operational requirements. When designing a forest, you have to consider an organization's organizational, operational, and legal requirements.

11. C. This scenario is a classic case of the need for the isolation model within a unique forest. If users are using a network application that could compromise the rest of your infrastructure, it's best to completely isolate these users. Thus, it's best to create a separate forest with a completely separate group inside an individual domain.

12. D. In a situation like this, the most important criteria are that the domain controllers be read-only, so that the information cannot be read if compromised, and that the hard drive's data be secured with BitLocker. This way, even if the hard drive is removed, the contents will not be useful to an intruder.

13. D. C is a very tempting option for this design but ultimately incorrect. The reason for this is that C would be designed using the isolation model, which isn't necessarily required for this design. Instead, you can use the autonomous model and still maintain control of your enterprise while implementing the required Windows legacy NT support and keeping overall control of the network.

14. A. Active Directory Rights Management Services is an administrative tool that allows administrators to enable users to assign specific rights to their files. To enable this feature, you must manually set it up and have access to a database server, such as Microsoft SQL Server 2008.

15. B. The best way to connect two sites that already have existing site links is to use a site-link bridge. By default, when you're creating sites, the Bridge All Site Links option is enabled. Thus, it's recommended you keep this enabled so you'll be able to communicate through preexisting sites.

16. C. If you have less than 500 users, Microsoft recommends at least 512MB of RAM and a single processor system. Above that, the minimum requirements are outlined in the tables in this chapter.

17. A. An organizational forest divides the administrative structure of its users by departments, products, or other definite separations of duties based upon the business model of the environment. In this case, the topology of the forest follows the structure of the organization.

18. A. An organizational forest divides the administrative structure of its users by departments, products, or other definite separations of duties based upon the business model of the environment. In this case, the topology of the forest follows the structure of the organization.

19. B. A shortcut trust would be best for this situation. This is because without it, the server would have to authenticate through five other domains in order to receive its credentials. With a shortcut trust, the need for this extra level of administration is removed.

20. A. Because domain C is a Windows NT domain, an external trust will be required. With pre–Active Directory domain controllers, this is the only way to properly authenticate.

21. C. To authenticate properly, a Unix server requires a realm trust to be created. This is so the standard Unix directory structure can communicate effectively with the Windows Active Directory.

22. A, E. The primary responsibility of a domain controller is the authentication and authorization of users. A domain controller logs in users by validating their credentials and then allowing them to access network resources.