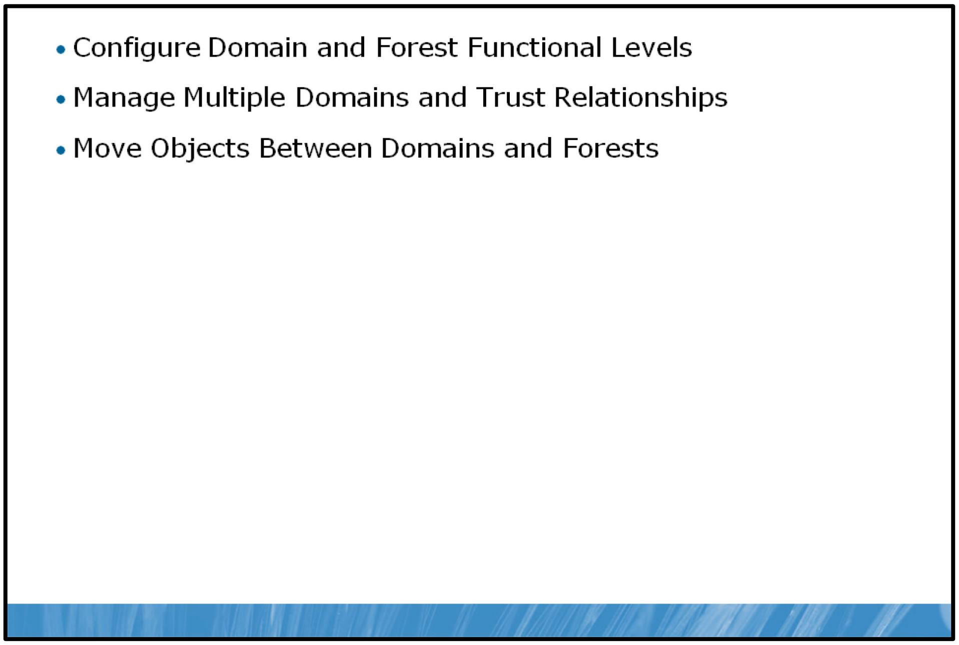
Module 15

Managing Multiple Domains and Forests

Contents:

Module Overview



- Configure Domain and Forest Functional Levels
- Manage Multiple Domains and Trust Relationships
- Move Objects Between Domains and Forests

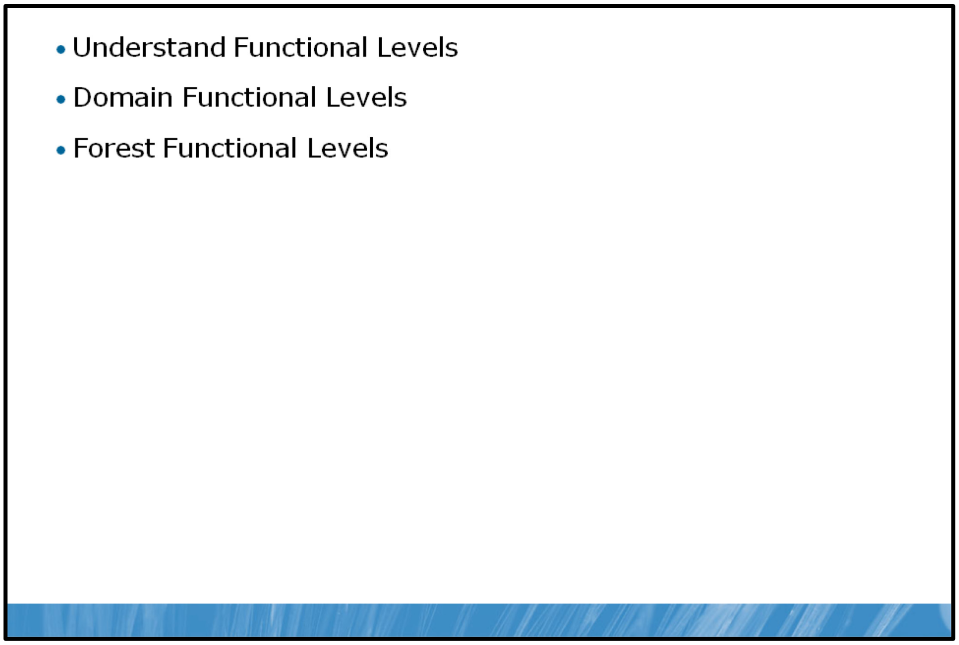
In Module 1, you learned that Active Directory Domain Services (AD DS) provides the foundation for an identity and access management solution, and you explored the creation of a simple AD DS infrastructure consisting of a single forest and a single domain. In subsequent modules, you understood the details of managing an AD DS environment. In this module, you will explore the highest level of an AD DS infrastructure and consider the model and functionality of your domains and forests. You will learn how to raise the domain and forest functionality levels within your environment, how to design the optimal AD DS infrastructure for your enterprise, how to migrate objects between domains and forests, and how to enable authentication and resources access across multiple domains and forests.

Objectives

After completing this module, you will be able to:

- Configure domain and forest functional levels.
- Manage multiple domains and trust relationships.
- Move objects between domains and forests.

Lesson 1
Configure Domain and Forest Functional Levels



- Understand Functional Levels
- Domain Functional Levels
- Forest Functional Levels

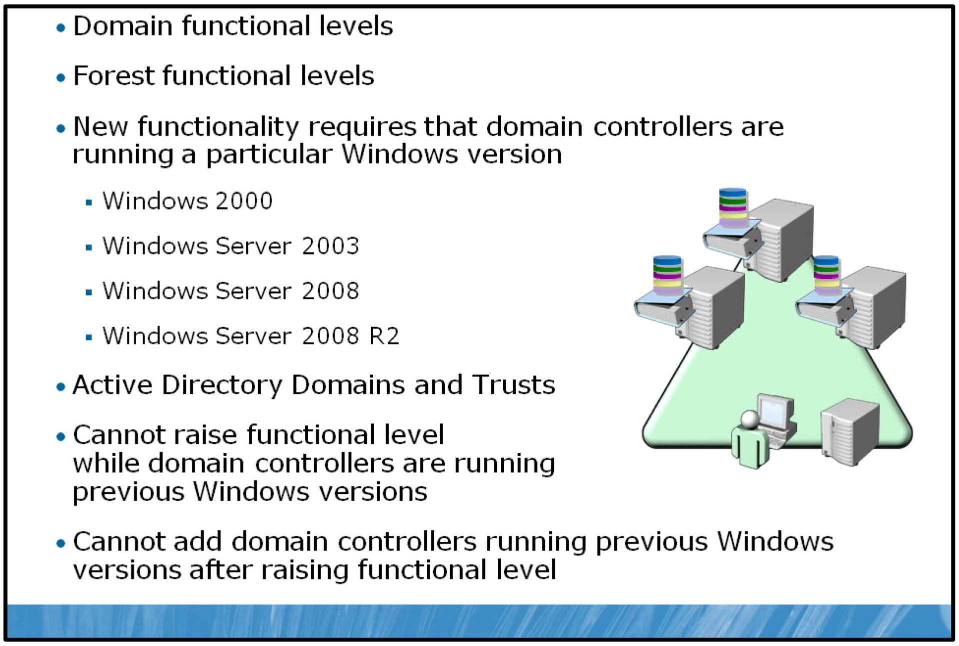
When you implement Windows Server® 2008 domain controllers in your domains and forest, you can begin to take advantage of new capabilities in AD DS. Domain and forest functional levels are operating modes of domains and forests, respectively. Functional levels determine the Windows® versions that you can use as domain controllers and the availability of Active Directory features.

Objectives

After completing this lesson, you will be able to:

- Understand domain and forest functional levels.
- Raise domain and forest functional levels.
- Identify capabilities added by each functional level.

Understand Functional Levels



- Domain functional levels
- Forest functional levels
- New functionality requires that domain controllers are running a particular Windows version
  - Windows 2000
  - Windows Server 2003
  - Windows Server 2008
  - Windows Server 2008 R2
- Active Directory Domains and Trusts
- Cannot raise functional level while domain controllers are running previous Windows versions
- Cannot add domain controllers running previous Windows versions after raising functional level

Functional levels are like switches that enable new functionality offered by each version of Windows. Windows Server 2003 added several features to Active Directory, and Windows Server 2008, and Windows Server 2008 R2 continues the evolution of AD DS. These features are not backward-compatible, so if you have domain controllers running Windows 2000 Server, you cannot enable the functionality offered by later versions of Windows, and the newer functionality is disabled. Similarly, until all domain controllers are running Windows Server 2008, you cannot implement its enhancements to AD DS. Raising the functional level entails two major requirements:

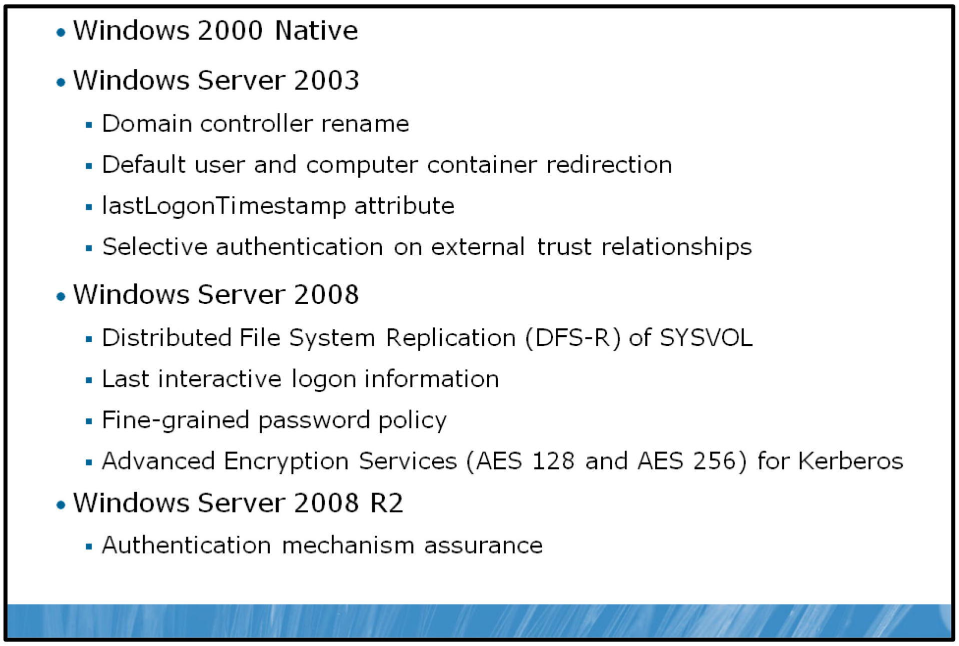
1. All domain controllers must be running the correct version of Windows Server.
2. You must manually raise the functional level. It does not happen automatically.

Remember that only domain controllers determine your ability to set a functional level. You can have member servers and workstations running any version of Windows within a domain or forest at any functional level.

It's important to note that raising a functional level is a one-way operation: you cannot lower a domain or forest functional level. Therefore, after you have raised the domain functional level to Windows Server 2008, for example, you cannot at a later date add a domain controller running at Windows Server 2003 to the same domain.

It's also important to note that a forest can have domains running at different functional levels, but after the forest functional level has been raised, you cannot add a domain controller running a lower version of Windows to any domain in the forest.

Domain Functional Levels



- Windows 2000 Native
- Windows Server 2003
  - Domain controller rename
  - Default user and computer container redirection
  - lastLogonTimestamp attribute
  - Selective authentication on external trust relationships
- Windows Server 2008
  - Distributed File System Replication (DFS-R) of SYSVOL
  - Last interactive logon information
  - Fine-grained password policy
  - Advanced Encryption Services (AES 128 and AES 256) for Kerberos
- Windows Server 2008 R2
  - Authentication mechanism assurance

The domain functional level affects the Active Directory features available within the domain and determines the Windows versions that are supported for domain controllers within the domain. In previous Windows versions, domain functional levels and modes, as they were called in Windows 2000 Server, supported domain controllers running Windows NT® 4.0. Support for Windows NT has ended with Windows Server 2008. All domain controllers must be running Windows 2000 Server or newer before you can add the first Windows Server 2008 domain controller to the domain. Windows Server 2008 Active Directory supports the following four domain functional levels:

- Windows 2000 Native
- Windows Server 2003
- Windows Server 2008
- Windows Server 2008 R2

Windows 2000 Native

The Windows 2000 Native domain functional level is the lowest functional level that supports a Windows Server 2008 domain controller. The following operating systems are supported for domain controllers:

- Windows 2000 Server
- Windows Server 2003
- Windows Server 2008
- Windows Server 2008 R2

If you have domain controllers running Windows 2000 Server or Windows Server 2003, or if you expect that you might add one or more domain controllers running those previous versions of Windows, you should leave the domain at the Windows 2000 Native functional level.

Windows Server 2003

After you have removed or upgraded all domain controllers running Windows 2000 Server, the domain functional level can be raised to Windows Server 2003. At this functional level, the domain can no longer support domain controllers running Windows 2000 Server, so all domain controllers must be running one of the following operating systems.

- Windows Server 2003
- Windows Server 2008
- Windows Server 2008 R2

The Windows Server 2003 domain functional level adds several new features to those offered at the Windows 2000 Native domain functional level. These features include the following:

- Domain controller rename. The domain management tool, netdom.exe, can be used to prepare for domain controller rename.
- The lastLogonTimestamp attribute. When a user or computer logs on to the domain, the lastLogonTimestamp attribute is updated with the logon time. This attribute is replicated within the domain.
- The userPassword attribute. Security principals in Active Directory include users, computers, and groups. A fourth object class, inetOrgPerson, is similar to a user and is used to integrate with several non-Microsoft directory services. At the Windows Server 2003 domain functional level, you can set the userPassword attribute as the effective password on both inetOrgPerson and user objects. This attribute is write-only. You cannot retrieve the password from the userPassword attribute.
- Default user and computer container redirection. In Module 5, you learned that you can use the redirusr.exe and redircmp.exe commands to redirect the default user and computer containers. This causes new accounts to be created in specific organizational units rather than in the Users and Computers containers.
- Authorization Manager policies. Authorization Manager, a tool that can be used to provide authorization by applications, can store its authorization policies in AD DS.
- Constrained delegation. Applications can take advantage of the secure delegation of user credentials through the Kerberos authentication protocol. You can configure delegation to be allowed only to specific destination services.
- Selective authentication. In Lesson 2 of this module, you will learn to create trust relationships between your domain and another domain or forest. Selective authentication enables you to specify the users and groups from the trusted domain or forest who are allowed to authenticate to servers in your forest.
- Read only domain controllers (RODCs). A domain must be at the Windows Server 2003 domain functional level before an RODC can be added. In addition, you must run adprep /rodcprep, and at least one writable Windows Server 2003 domain controller must be in place.

Windows Server 2008

When all domain controllers are running Windows Server 2008, and you are confident that you will not need to add domain controllers running previous versions of Windows, you can raise the domain functional level to Windows Server 2008.

The Windows Server 2008 domain functional level supports domain controllers running the following operating systems.

- Windows Server 2008
- Windows Server 2008 R2

The Windows Server 2008 domain functional level adds four domain-wide features to AD DS:

- DFS-R replication of SYSVOL. In Module 12, you learned to configure SYSVOL so that it is replicated with Distributed File System (DFS-R) instead of File Replication Service (FRS). DFS-R provides a more robust and detailed replication of SYSVOL contents.
- Advanced Encryption Services. You can increase the security of authentication with Advanced Encryption Services (AES 128 and AES 256) support for the Kerberos protocol. AES replaces the RC4-HMAC (Hash Message Authentication Code) encryption algorithm.
- Last interactive logon information. When a user logs on to the domain, several attributes of the user object are updated with the time, the workstation to which the user logged on, and the number of failed logon attempts since the last logon.
- Fine-grained password policies. In Module 10, you learned about fine-grained password policies, which enable you to specify unique password policies for users or groups in the domain.

Windows Server 2008 R2

The Windows Server 2008 R2 domain functional level supports domain controllers running only the following operating system:

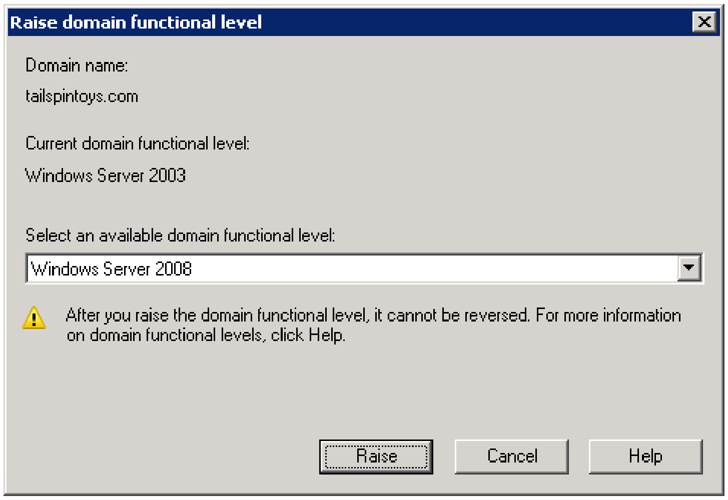
- Windows Server 2008 R2

The Windows Server 2008 R2 domain functional level adds the following domain-wide feature to AD DS:

- Authentication mechanism assurance. When this feature is enabled, additional information is provided within each user's Kerberos token. This information includes the type of logon method used, such as a smart card or a user name and password combination. Authentication mechanism assurance is used with the Active Directory Federation Services (AD FS) role to assist with authentication to claims-aware applications.

Raising the Domain Functional Level

You can raise the domain functional level after all domain controllers are running a supported version of Windows and when you are confident you will not have to add domain controllers running unsupported versions of Windows. To raise the domain functional level, open the Active Directory Domains and Trusts snap-in, right-click the domain, and select Raise Domain Functional Level. The dialog box shown here enables you to select a higher domain functional level.



Raise domain functional level

Domain name:

tailspintoys.com

Current domain functional level:

Windows Server 2003

Select an available domain functional level:

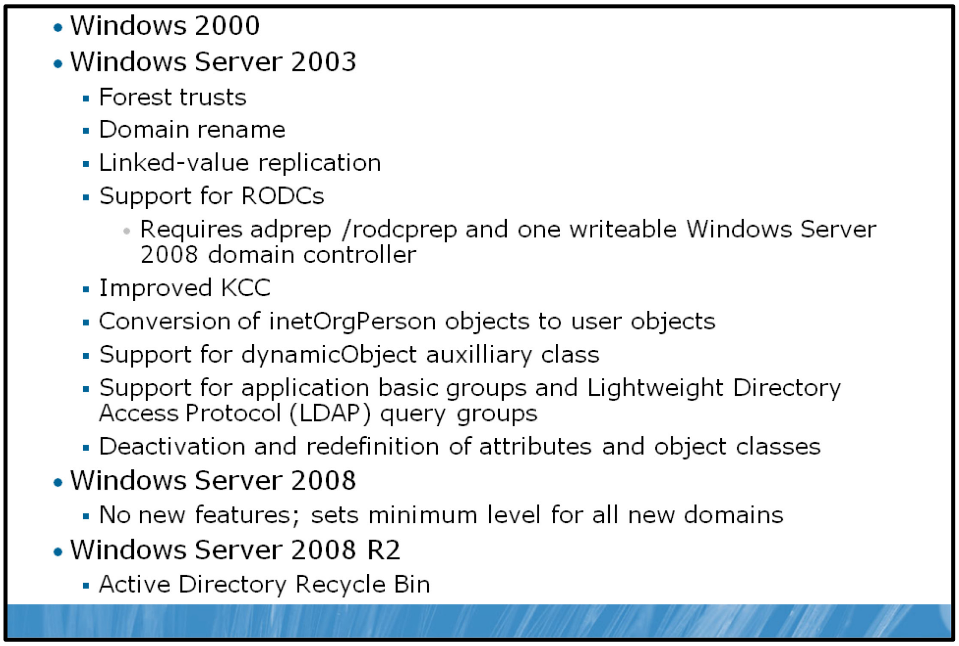
Windows Server 2008

After you raise the domain functional level, it cannot be reversed. For more information on domain functional levels, click Help.

Note Raising the domain functional level is a one-way operation. You cannot roll back to a previous domain functional level.

You can also raise the domain functional level by using the Active Directory Users and Computers snap-in. Right-click the domain and click Raise Domain Functional Level or right-click the root node of the snap-in and select Raise Domain Functional Level from the All Tasks menu.

Forest Functional Levels



- Windows 2000
- Windows Server 2003
  - Forest trusts
  - Domain rename
  - Linked-value replication
  - Support for RODCs
    - Requires adprep /rodcprep and one writeable Windows Server 2008 domain controller
  - Improved KCC
  - Conversion of inetOrgPerson objects to user objects
  - Support for dynamicObject auxilliary class
  - Support for application basic groups and Lightweight Directory Access Protocol (LDAP) query groups
  - Deactivation and redefinition of attributes and object classes
- Windows Server 2008
  - No new features; sets minimum level for all new domains
- Windows Server 2008 R2
  - Active Directory Recycle Bin

Just as domain functional levels enable certain domain-wide functionality and determine the Windows versions that are supported for domain controllers in the domain, forest functional levels enable forest-wide functionality and determine the operating systems supported for domain controllers in the entire forest. Windows Server 2008 R2 Active Directory supports three forest functional levels.

- Windows 2000
- Windows Server 2003
- Windows Server 2008
- Windows Server 2008 R2

Each functional level is described in the following sections.

Windows 2000

The Windows 2000 forest functional level is the baseline, default functional level. At the Windows 2000 functional level, domains can be running at any supported domain functional level.

- Windows 2000
- Windows Server 2003
- Windows Server 2008
- Windows Server 2008 R2

You can raise the forest functional level after all domains in the forest have been raised to the equivalent domain functional level.

Windows Server 2003

After all domains in the forest are at the Windows Server 2003 domain functional level, and when you do not expect to add any new domains with Windows 2000 Server domain controllers, you can raise the forest functional level to Windows Server 2003. At this forest functional level, domains can be running at the following domain functional levels.

- Windows Server 2003
- Windows Server 2008
- Windows Server 2008 R2

The following features are enabled at the Windows Server 2003 forest functional level:

- Forest trusts. In Lesson 2, you will learn to create trust relationships between forests.
- Domain rename. You can rename a domain within a forest.
- Linked-value replication. At the Windows 2000 forest functional level, a change to a group's membership results in the replication of the entire multivalued member attribute of the group. This can lead to increased replication traffic on the network and the potential loss of membership updates when a group is changed concurrently at different domain controllers. It also leads to a recommended cap of 5,000 members in any one group. Linked-value replication, enabled at the Windows Server 2003 forest functional level, replicates an individual membership change rather than the entire member attribute. This uses less bandwidth and prevents you from losing updates when a group is changed concurrently at different domain controllers.
- Support for RODCs. Module 10 discussed RODCs. RODCs are supported at the Windows Server 2003 forest functional level. Of course, the RODC itself must be running Windows Server 2008.
- Improved Knowledge Consistency Checker (KCC) algorithms and scalability. The intersite topology generator (ISTG) uses algorithms that enable AD DS to support replication in forests with more than 100 sites. At the Windows 2000 forest functional level, you must manually intervene to create replication topologies for forests with hundreds of sites. Additionally, the election of the ISTG uses an algorithm that is more efficient than at the Windows 2000 forest functional level.
- Conversion of inetOrgPerson objects to user objects. You can convert an instance of an inetOrgPerson object, used for compatibility with certain non-Microsoft directory services, into an instance of class user. You can also convert a user object to an inetOrgPerson object.
- Support for dynamicObject auxiliary class. The schema allows instances of the dynamic auxiliary class in domain directory partitions. This object class can be used by certain applications and by developers.
- Support for application basic groups and LDAP query groups. Two new group types, called application basic groups and LDAP query groups, can be used to support role-based authorization in applications that use Authorization Manager.
- Deactivation and redefinition of attributes and object classes. Although you cannot delete an attribute or object class in the schema at the Windows Server 2003 functional level, you can deactivate or redefine attributes or object classes.

Windows Server 2008

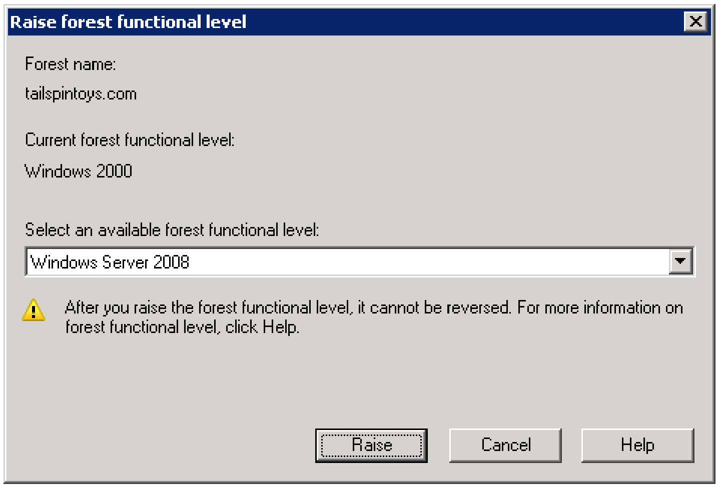
The Windows Server 2008 forest functional level does not add new forest-wide features. However, after the forest is configured to the Windows Server 2008 forest functional level, new domains added to the forest will operate at Windows Server 2008 domain functional level by default. At this forest functional level, all domains must be at the Windows Server 2008 domain functional level, which means that all domain controllers must be running Windows Server 2008.

Windows Server 2008 R2

The Windows Server 2008 R2 forest functional level adds the Active Directory Recycle Bin feature. This feature allows the ability to restore deleted Active Directory objects.

Raising the Forest Functional Level

Use the Active Directory Domains and Trusts snap-in to raise the forest functional level. Right-click the root node of the snap-in and Active Directory Domains and Trusts and click **Raise Forest Functional Level**. The dialog box shown here enables you to choose a higher forest functional level.



Raise forest functional level

Forest name:
tailspintoys.com

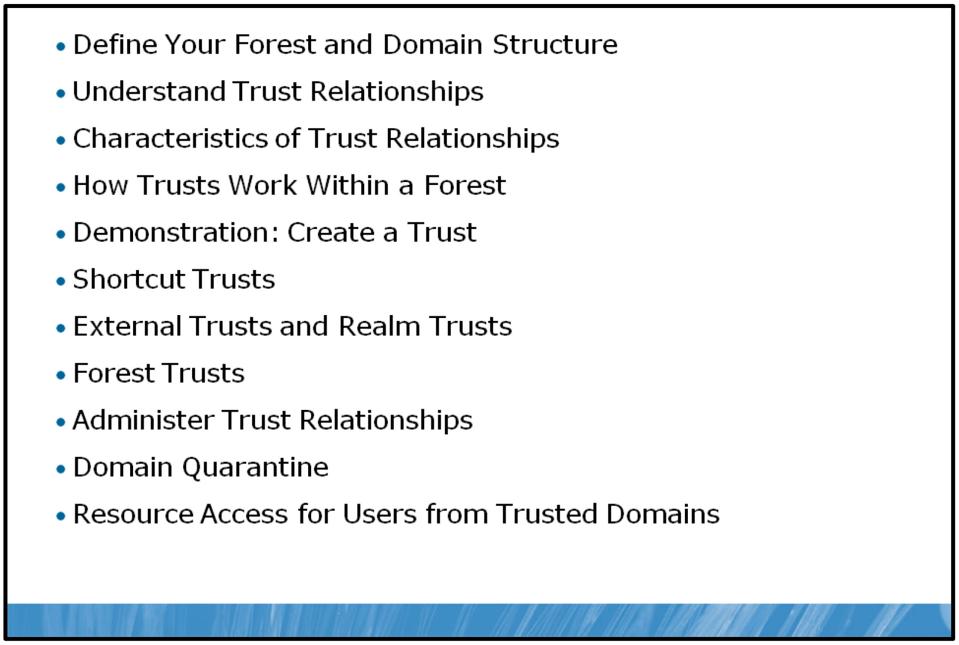
Current forest functional level:
Windows 2000

Select an available forest functional level:
Windows Server 2008

After you raise the forest functional level, it cannot be reversed. For more information on forest functional level, click Help.

Raise the forest functional level only when you are confident that you will not add new domains at unsupported domain functional levels. You cannot roll back to a previous forest functional level after raising it.

Lesson 2
Manage Multiple Domains and Trust Relationships



- Define Your Forest and Domain Structure
- Understand Trust Relationships
- Characteristics of Trust Relationships
- How Trusts Work Within a Forest
- Demonstration: Create a Trust
- Shortcut Trusts
- External Trusts and Realm Trusts
- Forest Trusts
- Administer Trust Relationships
- Domain Quarantine
- Resource Access for Users from Trusted Domains

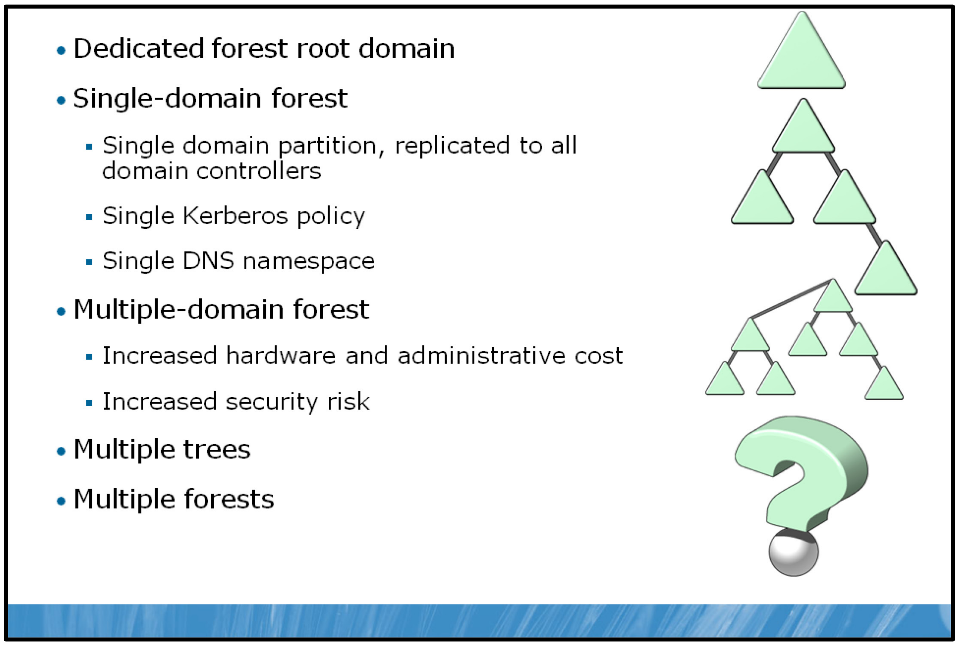
In previous modules, you learned how to configure, administer, and manage a single domain. However, your enterprise's Active Directory infrastructure might include a multi-domain forest or even more than one forest. You might need to move objects between domains or restructure your domain model entirely. You might also encounter requirements to enable authentication and access to resources across domains and forests. In this lesson, you will learn the skills required to support multiple domains and forests.

Objectives

After completing this lesson, you will be able to:

- Design an effective domain and tree structure for AD DS.
- Understand trust relationships.
- Configure, administer, and secure trust relationships.
- Identify the role of the Active Directory Migration Tool and the issues related to object migration and domain restructure.

Define Your Forest and Domain Structure



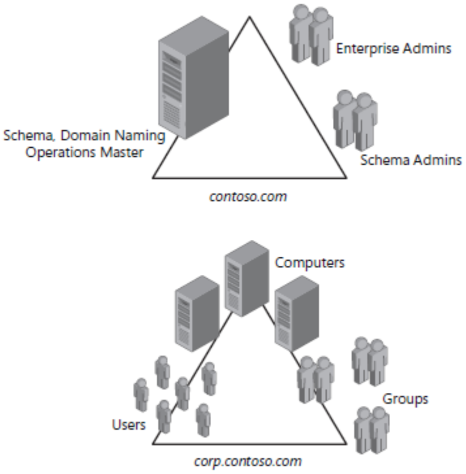
- Dedicated forest root domain
- Single-domain forest
  - Single domain partition, replicated to all domain controllers
  - Single Kerberos policy
  - Single DNS namespace
- Multiple-domain forest
  - Increased hardware and administrative cost
  - Increased security risk
- Multiple trees
- Multiple forests

With the perspective you have gained from the previous modules, you can now consider the design of your Active Directory forest, trees, and domains. Interestingly, the best practices guidance regarding forest and domain structure has evolved as enterprises around the world have put Active Directory into production in every conceivable configuration and as the Active Directory feature set has grown.

Dedicated Forest Root Domain

In earlier Active Directory versions, the recommendation was to create a dedicated forest root domain. You'll recall from Module 1 that the forest root domain is the first domain in the forest. A dedicated forest root domain's exclusive purpose is to administer the forest infrastructure. It contains, by default, the single master operations for the forest. It also contains highly sensitive groups, such as Enterprise Admins and Schema Admins that can have a far-reaching impact on the forest. The theory was that the dedicated forest root enhances the security around these forest-wide functions. The dedicated forest root domain would also be less likely to become obsolete and provide easier transfer of ownership. Underneath the dedicated forest root, according to early recommendations, is a single global child domain with all the objects one thinks of in a domain: users, groups, computers, and so on.

The structure would look something like the following figure.



Note Implementation of a dedicated forest root domain may be beneficial for larger enterprises. A single domain forest is the most common design for small to medium organizations. There is no single design that is appropriate for every organization, so you must examine the characteristics of your enterprise against the design criteria presented later in this lesson.

Single-Domain Forest

For many organizations, building a forest with a single domain is common. The experience and knowledge that have led to this option include the following points.

- There are risks and costs associated with any multidomain forest, as you'll learn later in this lesson. A single domain bears the lowest hardware and support cost and reduces certain risks.
- There are not yet tools that enable an enterprise to perform pruning and grafting of Active Directory trees. In other words, you cannot break a domain from your tree and transplant it in the forest of another enterprise. If that were possible, a dedicated forest root that you could maintain while transferring domains in and out of your forest would make more sense.
- You can implement least-privilege security within a single domain that is at least as secure as, if not more secure than, security in a forest with a dedicated forest root and a child domain.

Therefore, when you consider your domain design, you should begin with the assumption that you will have a single domain in your forest.

Multiple Domain Forest

In some scenarios, a multiple-domain forest is required. The important point to remember is that you should never create a multiple-domain forest simply to reflect the organizational structure of your business. That structure—business units, divisions, departments, and offices—will change over time. The logical structure of your directory service should not be dependent solely on organizational characteristics.

Instead, your domain model should be derived from the characteristics of domains themselves. There are certain properties of a domain that affect all objects within the domain, and if that consistent effect is not appropriate for your business requirements, you must create additional domains. A domain is characterized by the following.

- A single domain partition, replicated to all domain controllers. The domain naming context contains the objects for users, computers, groups, policies, and other domain resources. It is replicated to every domain controller in the domain. If you need to partition replication for network topology considerations, you must create separate domains. Consider, however, that Active Directory replication is extremely efficient and can support large domains over connections with minimal bandwidth.
- If there are legal or business requirements that restrict replication of certain data to locations where you maintain domain controllers, you need to either avoid storing that data in the domain partition or create separate domains to segregate replication. In such cases, you should also ensure that the global catalog is not replicating that data.
- Because legal and technical issues surrounding replication tend to affect the global catalog and potentially other data stores. Organizations with these concerns are increasingly turning to multiple forest models.
- A single Kerberos policy. The default Kerberos policy settings in AD DS are sufficient for most enterprises. If, however, you need distinct Kerberos policies, you will require distinct domains.
- A single DNS namespace. An Active Directory domain has a single DNS domain name. If you need multiple domain names, you would need multiple domains. However, consider the costs and risks of multiple domains before modeling your directory service domains to match arbitrary DNS name requirements.

In domains running domain functional levels lower than Windows Server 2008, a domain can support only one password and account lockout policy. Therefore, in the earlier versions of Windows, an organization requiring multiple password policies would need multiple domains to support that requirement. This is no longer the case in Windows Server 2008 or Windows Server 2008 R2, which, at the appropriate domain functional level, can support fine-grained password policies.

Adding domains to a forest increases administrative and hardware costs. Each domain must be supported by at least two domain controllers, which must be backed up, secured, and managed. Even more domain controllers might be required to support cross-domain resource access in a geographically distributed enterprise. When you add more domains, you might need to move users between domains. This is more complicated than moving users between organizational units (OUs). Group Policy objects and access control settings that are common for the enterprise will have to be duplicated for each domain.

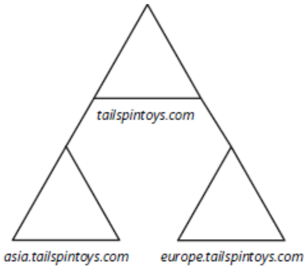
These are just a few of the costs associated with a multiple-domain environment. There are also security risks involved with having multiple domains. Most of these risks relate to the fact that a domain is not a security boundary—a forest is the security boundary. Within a forest, service administrators can cause forest-wide damage. There are several categories of vulnerability whereby a compromised administrative account, or an administrator with bad intent, could cause denial of service or damage to the forest integrity.

For example, an administrator in any domain can create universal groups, the membership of which is replicated to the GC. By creating multiple universal groups and overpopulating the member attribute, excessive replication could lead to denial of service on domain controllers acting as domain controllers in other domains. An administrator in any domain could also restore an outdated backup of the directory, which could corrupt the forest.

In a multidomain forest, it might make sense to create a dedicated forest root domain as an empty domain to act as the trust root for the forest. Trust roots will be discussed later in this lesson.

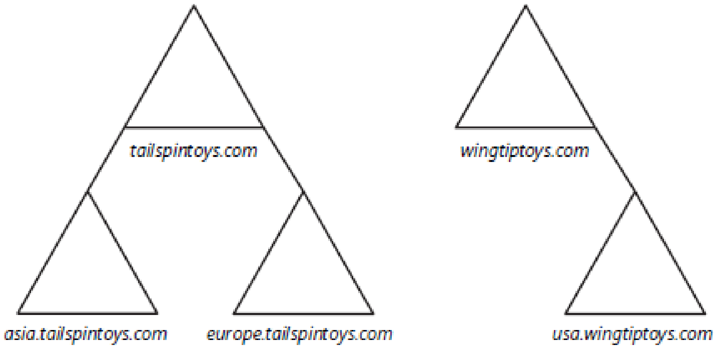
Multiple Trees

Remember that a tree is defined as a contiguous DNS namespace. If you have more than one domain, you can decide whether those domains share a contiguous DNS namespace and form a single tree, as shown in the first figure here, or are in a noncontiguous DNS namespace, forming multiple trees, as shown in the second figure.



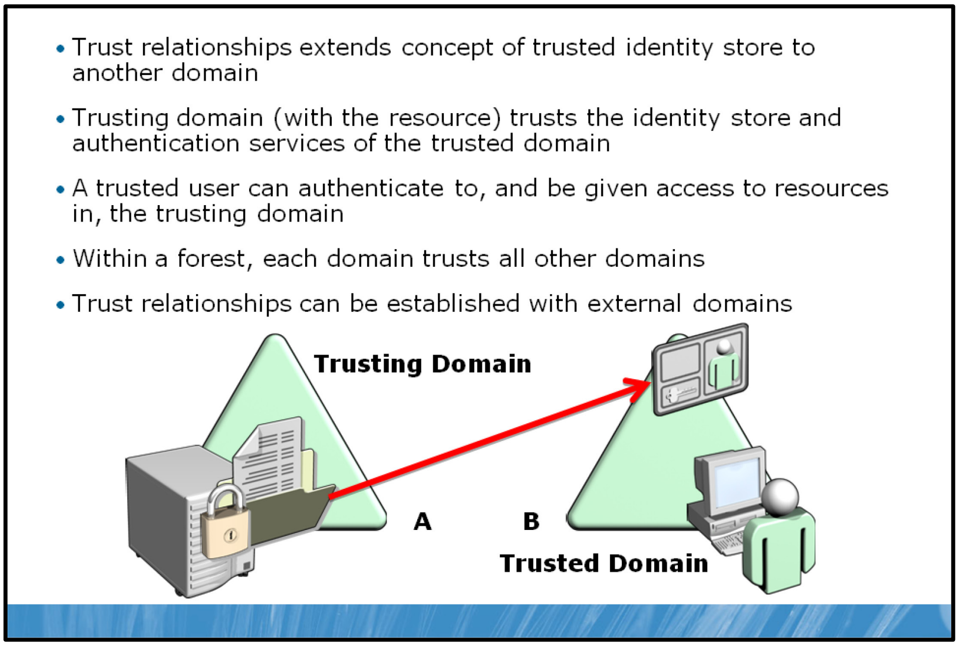
Multiple Forests

A forest is an instance of Active Directory. All domains and domain controllers in a forest share replicas of the schema and configuration. Domain controllers that are global catalog servers host partial attribute sets for all objects in other domains in the forest. Domains in a forest share transitive, two-way trusts, meaning that all users in the domain belong to the Authenticated Users special identity in every domain. The forest's Enterprise Admins, Schema Admins, and Administrators groups in the forest root domain wield significant power over all objects in the forest.



If any of these characteristics of a forest are at odds with your business requirements, you might need multiple forests. In fact, given the market's current concerns with security, many consultants are recommending that organizations design either a single-domain forest or use multiple forests. Cross-forest trusts, discussed later in this lesson, and Active Directory Federation Services (AD FS) make it easier to manage authentication in multiple-forest enterprises.

Understand Trust Relationships



- Trust relationships extends concept of trusted identity store to another domain
- Trusting domain (with the resource) trusts the identity store and authentication services of the trusted domain
- A trusted user can authenticate to, and be given access to resources in, the trusting domain
- Within a forest, each domain trusts all other domains
- Trust relationships can be established with external domains

Whenever you are implementing a scenario involving two or more AD DS domains, it is likely that you will be working with trust relationships, or trusts. It is important that you understand the purpose, functionality, and configuration of trust relationships.

Trust Relationships Within a Domain

In Module 1, you were guided through what happens when a domain member server or workstation joins a domain. While in a workgroup, the computer maintains an identity store in the security accounts manager (SAM) database, and it authenticates users against that identity store and secures system resources only with identities from the SAM database. When the computer joins a domain, it forms a trust relationship with the domain. The effect of that trust is that the computer allows users to be authenticated not by the local system and its local identity store, but by the authentication services and identity store of the domain: AD DS. The domain member also allows domain identities to be used to secure system resources. For example, Domain Users is added to the local Users group, giving Domain Users the right to log on locally to the system. Also, domain user and group accounts can be added to access control lists (ACLs) on files, folders, registry keys, and printers on the system. All domain members have similar trust relationships with the domain, enabling the domain to be a central store of identity and a centralized service providing authentication.

Trust Relationships Between Domains

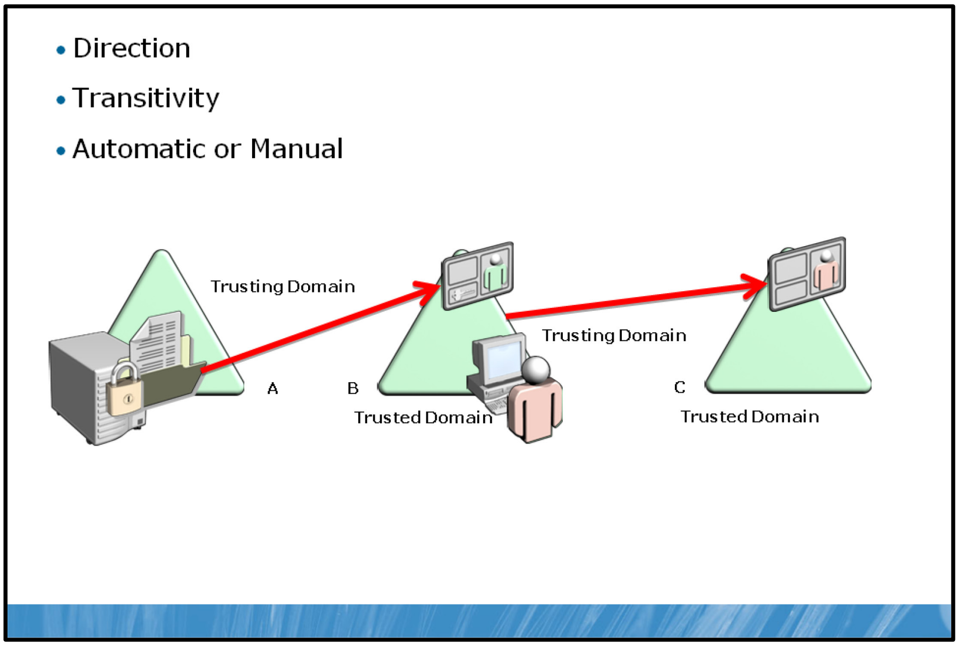
You can extend the concept of trust relationships to other domains. A trust relationship between two domains enables one domain to trust the authentication service and the identity store of another domain and to use those identities to secure resources. In effect, a trust relationship is a logical link established between domains to enable pass-through authentication.

There are two domains in every trust relationship: a trusting domain and a trusted domain. The trusted domain holds the identity store and provides authentication for users in that identity store. When a user in the directory of the trusted domain logs on to or connects to a system in the trusting domain, the trusting domain cannot authenticate that user because the user is not in its data store, so it passes the authentication to a domain controller in the trusted domain. The trusting domain, therefore, trusts the trusted domain to authenticate the identity of the user. The trusting domain extends trust to the authentication services and the identity store of the trusted domain.

Because the trusting domain trusts the identities in the trusted domain, the trusting domain can use the trusted identities to grant access to resources. Users in a trusted domain can be given user rights such as the right to log on to workstations in the trusting domain. Users or global groups in the trusted domain can be added to domain local groups in the trusting domain. Users or global groups in the trusted domain can be given permissions to shared folders by adding the identities to ACLs in the trusting domain.

The terminology can be confusing, and it is often easier to understand trust relationships when you look at an illustration. The diagram shown on the slide shows a simple trust relationship. Domain A trusts Domain B. That makes Domain A the trusting domain and Domain B the trusted domain. If a user in Domain B connects to or logs on to a computer in Domain A, Domain A will pass the authentication request to a domain controller in Domain B. Domain A can also use the identities from Domain B—users and groups, for example—to grant user rights and resource access in Domain A. A user or group in Domain B can, therefore, be added to an ACL on a shared folder in Domain A. A user or group in Domain B can also be added to a domain local group in Domain A.

Characteristics of Trust Relationships



- Direction
- Transitivity
- Automatic or Manual

Trust relationships between domains can be characterized by three attributes of the trust: direction, transitivity, and automatic or manual.

Direction

A trust relationship can be one-way or two-way. In a one-way trust, such as the trusts illustrated so far, users in the trusted domain can be given access to resources in the trusting domain, but users in the trusting domain cannot be given access to resources in the trusted domain. In most cases, you can create a second, one-way trust in the opposite direction to achieve that goal. For example, you can create a second trust relationship in which Domain B trusts Domain A. Some trust relationships are by nature two-way. In a two-way trust relationship, both domains trust the identities and authentication services of the other domain.

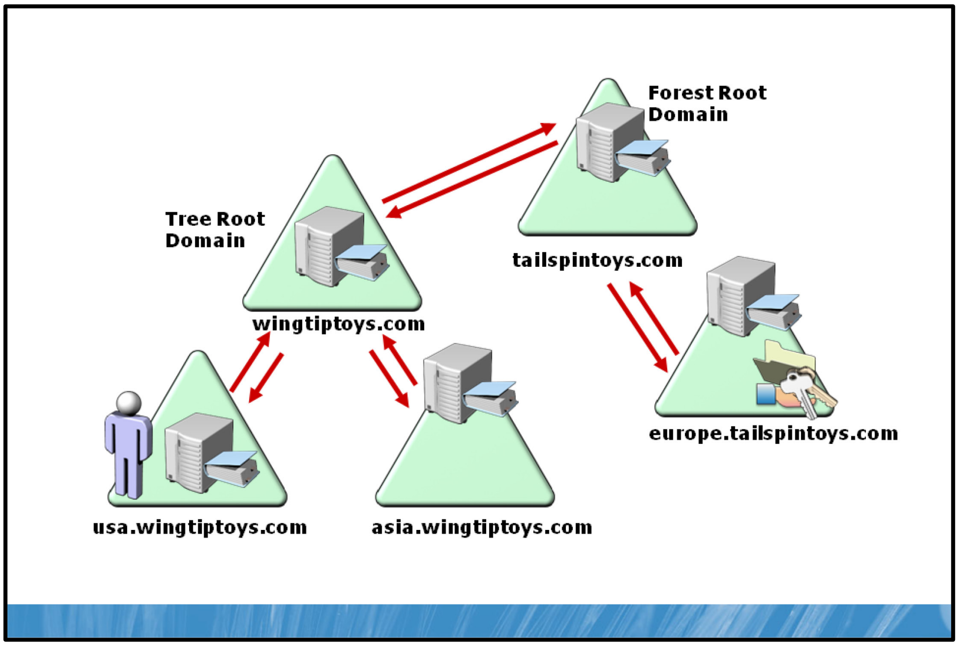
Transitivity

Some trusts are not transitive, and others are transitive. In the figure above, Domain A trusts Domain B, and Domain B trusts Domain C. If the trusts are transitive, Domain A trusts Domain C. If they are not transitive, Domain A does not trust Domain C. In most cases, you could create a third trust relationship, specifying that Domain A trusts Domain C. With transitive trusts, that third relationship is not necessary; it is implied.

Automatic or Manual

Some trusts are created automatically. Other trusts must be created manually.

How Trusts Work Within a Forest



Within a forest, all domains trust each other. That is because the root domain of each tree in a forest trusts the forest root domain—the first domain installed in the forest—and each child domain trusts its parent domain. All trusts automatically created should never be deleted and are transitive and two-way. The net result is that a domain trusts the identity stores and authentication services of all other domains in its forest. Users and global groups from any domain in the forest can be added to domain local groups, can be given user rights, and can be added to ACLs on resources in any other domain in the forest. Trusts to other forests and domains outside the forest must be manually established. With that summary, you can look at the details of trusts within and outside of an Active Directory forest.

Authentication Protocols

Windows Server 2008 Active Directory authenticates users with one of two protocols—Kerberos version 5 (v5) or NTLM. Kerberos v5 is the default protocol used by computers running Windows Server 2008, Windows Vista®, Windows Server 2003, Windows XP, and Windows 2000 Server. If a computer involved in an authentication transaction does not support Kerberos v5, the NTLM protocol is used instead. Group Policies can be used to disable NTLM authentication.

Kerberos Authentication Within a Domain

When a user logs on to a client running Kerberos v5, the authentication request is forwarded to a domain controller. Each Active Directory domain controller acts as a key distribution center (KDC), a core component of Kerberos. After validating the identity of the user, the KDC on the domain controller gives the authenticated user what is known as a ticket-granting ticket (TGT).

When the user needs to access resources on a computer in the same domain, the user must first obtain a valid session ticket for the computer. Session tickets are provided by the KDC of a domain controller, so the user returns to a domain controller to request a session ticket. The user presents the TGT as proof that he or she has already been authenticated. This enables the KDC to respond to the user's session ticket request without having to re-authenticate the user's identity. The user's session ticket request specifies the computer and the service the user wants to access. The KDC identifies that the service is in the same domain based on the service principal name (SPN) of the requested server. The KDC then provides the user a session ticket for the service.

The user then connects to the service and presents the session ticket. The server is able to determine that the ticket is valid and that the user has been authenticated by the domain. This happens through private keys; a topic that is beyond the scope of this lesson. The server, therefore, does not need to authenticate the user; it accepts the authentication and identity provided by the domain with which the computer has a trust relationship.

All these Kerberos transactions are handled by Windows clients and servers and are transparent to users themselves.

Kerberos Authentication Across Domains in a Forest

Each child domain in a forest trusts its parent domain with an automatic, two-way, transitive trust called a parent-child trust. The root domain of each tree in a domain trusts the forest root domain with an automatic, two-way, transitive trust called a tree-root trust.

These trust relationships create what is referred to as the trust path or trust flow in a forest. The trust path is easy to understand with a diagram, shown on the slide. The forest consists of two trees, the tailspintoys.com tree and the wingtiptoys.com tree. The tailspintoys.com domain is the forest root domain. The illustration indicates that the wingtiptoys.com tree root domain trusts the tailspintoys.com domain.

Kerberos authentication uses the trust path to provide a user in one domain a session ticket to a service in another domain. If a user in usa.wingtiptoys.com requests access to a shared folder on a server in europe.tailspintoys.com, the following transactions occur.

1. The user logs on to a computer in usa.wingtiptoys.com and is authenticated by a domain controller in usa.wingtiptoys.com through the authentication process described in the previous section. The user obtains a TGT for the domain controller in usa.wingtiptoys.com.

   The user wants to connect to a shared folder on a server in europe.tailspintoys.com.

2. The user contacts the KDC of a domain controller in usa.wingtiptoys.com to request a session ticket for the server in europe.tailspintoys.com.

3. The domain controller in usa.wingtiptoys.com identifies, based on the SPN, that the desired service resides in europe.tailspintoys.com, and not in the local domain.

   The job of the KDC is to act as a trusted intermediary between a client and a service. If the KDC cannot provide a session ticket for the service because the service is in a trusted domain and not in the local domain, the KDC provides the client a referral to help obtain the session ticket it is requesting.

   The KDC uses a simple algorithm to determine the next step. If the KDC domain is trusted directly by the service's domain, the KDC gives the client a referral to a domain controller in the service's domain. But if a transitive trust exists between the KDC and the service's domain, the KDC provides the client a referral to the next domain in the trust path.

4. The usa.wingtiptoys.com domain is not trusted directly by europe.tailspintoys.com; only a transitive trust exists between the two domains. Therefore, the KDC in the usa.wingtiptoys.com domain gives the client a referral to a domain controller in the next domain in the trust path, wingtiptoys.com.

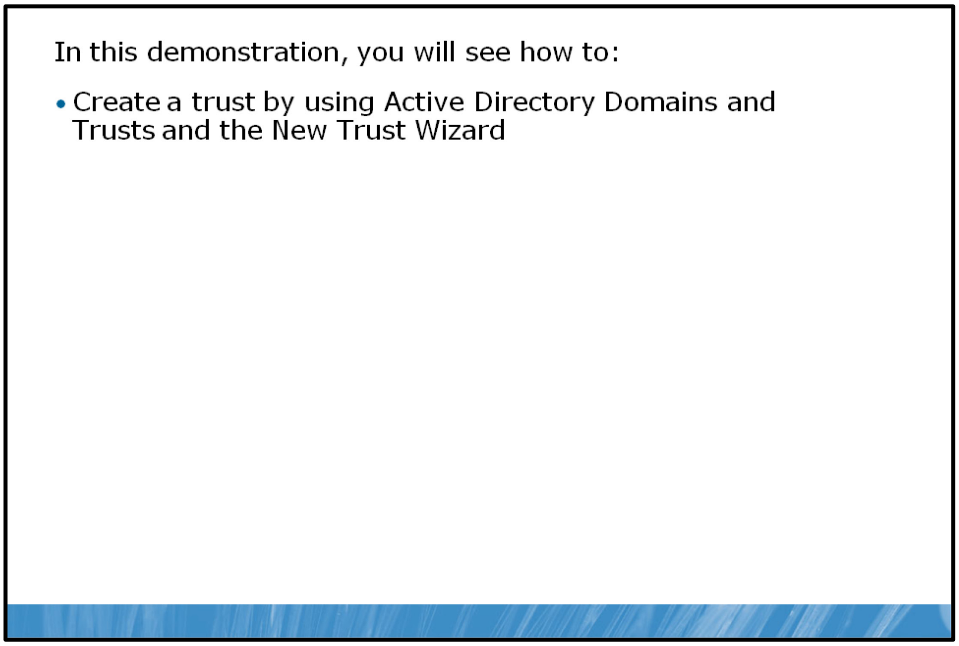
5. The client contacts the KDC in the referral domain, wingtiptoys.com.

6. Again, the KDC determines that the service is not in the local domain and that europe.tailspintoys.com does not trust wingtiptoys.com directly, and returns a referral to a domain controller in the next domain in the trust path, tailspintoys.com.
7. The client contacts the KDC in the referral domain, tailspintoys.com.
8. The KDC determines that the service is not in the local domain and that europe.tailspintoys.com truststailspintoys.com directly. Therefore, it returns a referral to a domain controller in the europe.tailspintoys.com domain.
9. The client contacts the KDC in the referral domain, europe.tailspintoys.com.
10. The KDC in europe.tailspintoys.com returns to the client a session ticket for the service.
11. The client contacts the server and provides the session ticket; the server provides access to the shared folder based on the permissions assigned to the user and the groups to which the user belongs.

    This process might seem complicated, but recall that it is handled in a way that is completely transparent to the user.

    The reverse process occurs if a user from usa.wingtiptoys.com logs on to a computer in the europe.tailspintoys.com domain. The initial authentication request must traverse the trust path to reach a KDC in the usa.wingtiptoys.com domain to authenticate the user.

Demonstration: Create a Trust



In this demonstration, you will see how to:

- Create a trust by using Active Directory Domains and Trusts and the New Trust Wizard

The steps for creating trusts are similar across categories of trusts. You must be a member of the Domain Admins or Enterprise Admins group to create a trust successfully.

To create a trust relationship:

1. Open the **Active Directory Domains and Trusts** snap-in.
2. Right-click the domain that will participate in one side of the trust relationship, and click **Properties**.

   You must be running Active Directory Domains and Trusts with credentials that have permissions to create trusts in this domain.
3. Click the **Trusts** tab.
4. Click the **New Trust** button.

   The New Trust Wizard guides you through the creation of the trust.
5. On the **Trust Name** page, type the DNS name of the other domain in the trust relationship, and then click **Next**.
6. If the domain you entered is not within the same forest, you will be prompted to select the type of trust, which will be one of the following:

   - **Forest**
   - **External**
   - **Realm**

   If the domain is in the same forest, the wizard knows it is a shortcut trust.
7. If you are creating a realm trust, you will be prompted to indicate whether the trust is transitive or nontransitive. (Realm trusts are discussed later in this lesson.)

8. On the **Direction Of Trust** page, select one of the following:
   - **Two-Way.** This establishes a two-way trust between the domains.
   - **One-Way: Incoming.** This establishes a one-way trust in which the domain you selected in step 2 is the trusted domain, and the domain you entered in step 5 is the trusting domain.
   - **One-Way: Outgoing.** This establishes a one-way trust in which the domain you selected in step 2 is the trusting domain, and a domain you entered in step 5 is the trusted domain.
9. Click **Next**.
10. On the **Sides Of Trust** page, select one of the following:
    - **Both this domain and the specified domain.** This establishes both sides of the trust. This requires that you have permission to create trusts in both domains.
    - **This domain Only.** This creates the trust relationship in the domain you selected in step 2. An administrator with permission to create trusts in the other domain must repeat this process to complete the trust relationship.
    - The next steps will depend on the options you selected in steps 8 and 10. The steps will involve one of the following:
    - If you selected **Both this domain and the specified domain**, you must enter a user name and password with permissions to create the trust in the domain specified in step 5.
    - If you selected **This Domain Only**, you must enter a trust password. A trust password is entered by administrators on each side of a trust to establish the trust. The passwords should not be the administrators' user account passwords. Instead, each should be a unique password used only for creating this trust. The passwords are used to establish the trust, and then the domains change them immediately.
11. If the trust is an outgoing trust, you are prompted to choose one of the following:
    - **Selective** Authentication
    - Domain-Wide Authentication or Forest-Wide Authentication, depending on whether the trust type is an external trust or a forest trust, respectively.
12. The New Trust Wizard summarizes your selections on the Trust Selections Complete page. Click **Next**.

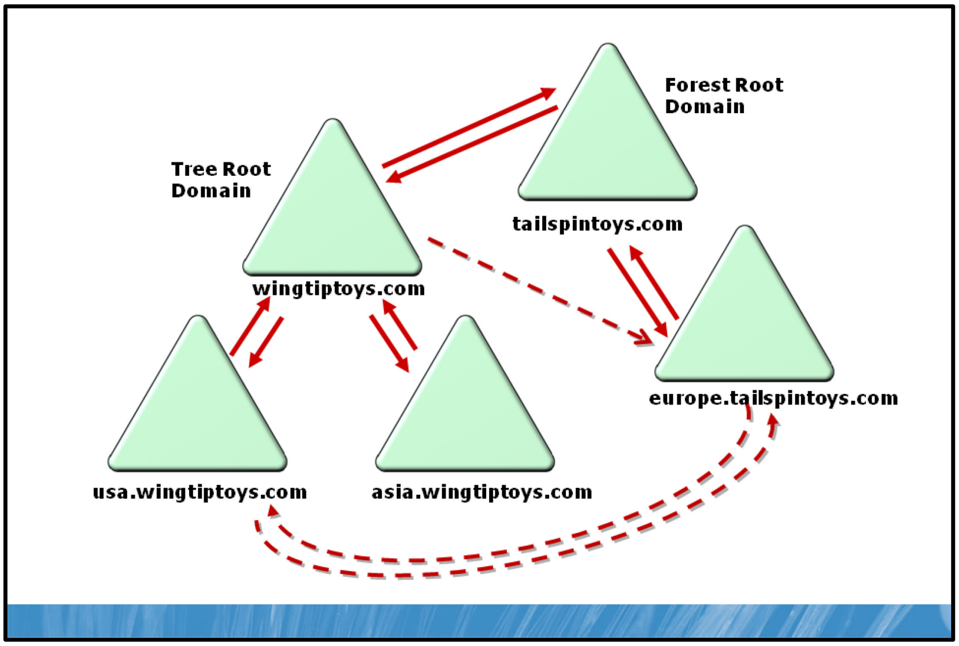
    The wizard creates the trust.
13. The **Trust Creation Complete** page appears. Verify the settings, and then click **Next**.

    You will then have the opportunity to confirm the trust. This option is useful if you have created both sides of the trust or if you are completing the second side of a trust.

    If you selected **Both this domain and the specified domain** in step 8, the process is complete. If you selected **This domain only** in step 8, the trust relationship will not be complete until an administrator in the other domain completes the process:
    - If the trust relationship you established is a one-way outgoing trust, an administrator in the other domain must create a one-way incoming trust.
    - If the trust relationship you established is a one-way incoming trust, an administrator in the other domain must create a one-way outgoing trust.

- If the trust relationship you established is a two-way trust, an administrator in the other domain must create a two-way trust.

Shortcut Trusts



The following four types of trusts must be created manually.

- Shortcut
- External
- Realm
- Forest

Each of these types of trusts will be discussed in the following sections.

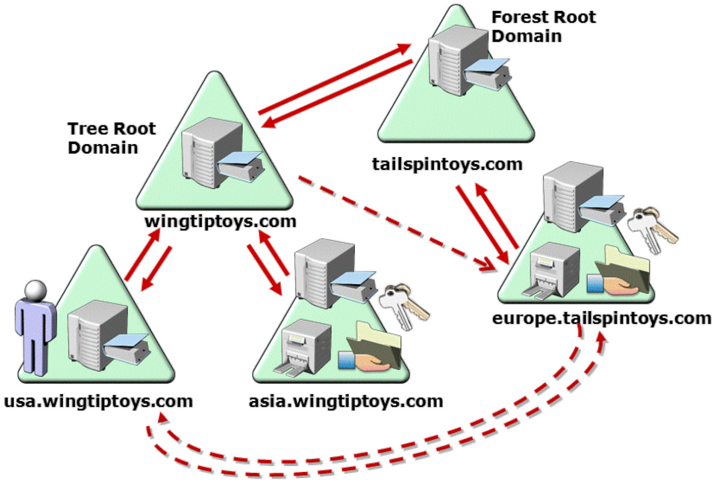
Shortcut Trusts

In an earlier section, there were the 11 steps of the process used to grant a session ticket for a client to access a resource in another domain within a forest. Most of those steps involved referrals to domains on the trust path between the user's domain and the domain of the shared folder. When a user from one domain logs on to a computer in another domain, the authentication request must also traverse the trust path. This can affect performance and, if a domain controller is not available in a domain along the trust path, the client will not be able to authenticate or to access the service.

Shortcut trusts are designed to overcome those problems by creating a trust relationship directly between child domains in the forest trust path.

Shortcut trusts optimize authentication and session ticket requests between domains in a multidomain forest. By eliminating the trust path, they eliminate the time required to traverse the trust path and thereby can significantly improve performance of session ticket requests.

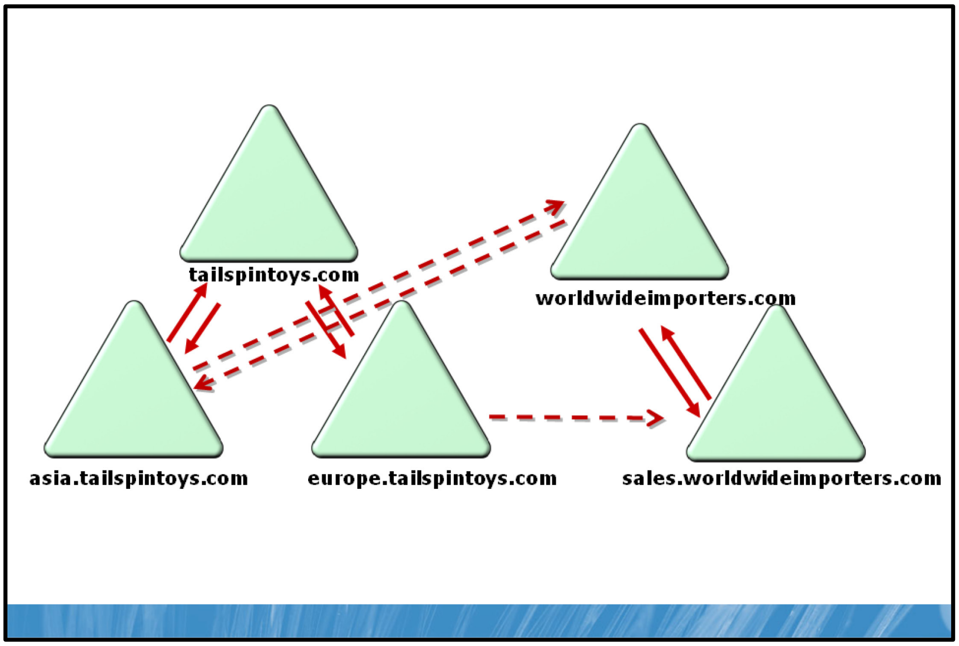
Shortcut trusts can be one-way or two-way. In either case, the trust is transitive. In the illustration on the slide, a one-way shortcut trust exists wherebywingtiptoys.com trusts the europe.tailspintoys.com.



When a user from europe.tailspintoys.com logs on to a computer in wingtiptoys.com or requests a resource in wingtiptoys.com, the request can be referred directly to a domain controller in the trusted domain, asia.wingitiptoys.com. However, the reverse is not true. If a user in wingtiptoys.com logs on to a computer in europe.tailspintoys.com, the authentication request will traverse the trust path up to tailspintoys.com and down to wingtiptoys.com.

A two-way shortcut trust is illustrated between usa.wingtiptoys.com and europe.tailspintoys.com. Users in both domains can be authenticated by and can request resources from computers in the other domain, and the shortcut trust path will be used.

External Trusts and Realm Trusts



External trusts and realm trusts are both used to provide resource access and authentication to other directory structures outside of your forest. The following sections discuss these two types of trusts.

External Trusts

When you need to work with a domain that is not in your forest, you might need to create an external trust. An external trust is a trust relationship between a domain in your forest and a Windows domain that is not in your forest. Examples are shown on the slide.

On the slide, you can see a one-way trust between the sales.worldwideimporters.com domain and the europe.tailspintoys.com domain. The Europe domain trusts the Sales domain, so users in the Sales domain can log on to computers in the Europe domain or connect to resources in the Europe domain.

The illustration shows a two-way trust between the worldwideimporters.com domain and the asia.tailspintoys.com domain. Users in each domain can be given access to resources in the other domain. Technically, all external trusts are nontransitive, one-way trusts. When you create a two-way external trust, you are actually creating two one-way trusts, one in each direction.

When you create an outgoing external trust, Active Directory creates a foreign security principal object for each security principal in the trusted domain. Those users, groups, and computers can then be added to domain local groups or ACLs on resources in the trusting domain.

To increase the security of an external trust relationship, you can choose Selective Authentication on the Outgoing Trust Authentication Level page of the New Trust Wizard. Additionally, domain quarantine, also called SID filtering, is enabled by default on all external trusts.

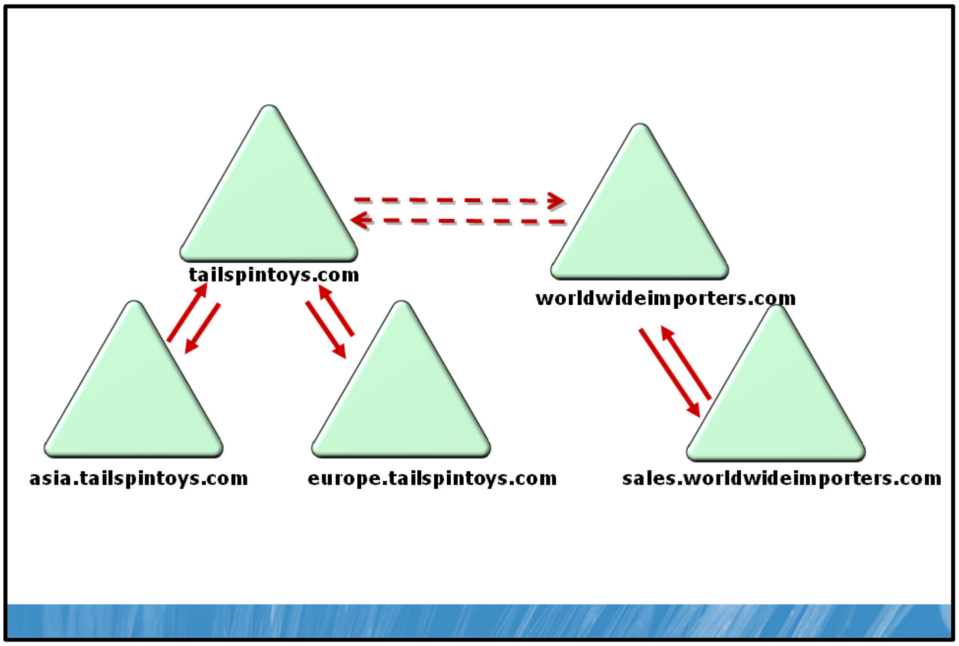
Realm Trusts

When you need cross-platform interoperability with security services based on other Kerberos v5 implementations, you can establish a realm trust between your domain and a UNIX Kerberos v5 realm.

Realm trusts are one-way, but you can establish one-way trusts in each direction to create a two-way trust. By default, realm trusts are nontransitive, but they can be made transitive.

If a non-Windows Kerberos v5 realm trusts your domain, the realm trusts all security principals in your domain. If your domain trusts a non-Windows Kerberos v5 realm, users in the realm can be given access to resources in your domain; however, the process is indirect. When users are authenticated by a non-Windows Kerberos realm, Kerberos tickets do not contain all the authorization data needed for Windows. Therefore, an account mapping system is used. Security principals are created in the Windows domain and are mapped to a foreign Kerberos identity in the trusted non-Windows Kerberos realm. The Windows domain uses only these proxy accounts to evaluate access to domain objects that have security descriptors. All Windows proxy accounts can be used in groups and on ACLs to control access on behalf of the non-Windows security principal. Account mappings are managed through Active Directory Users and Computers.

Forest Trusts



When you require collaboration between two separate organizations represented by two separate forests, you can consider implementing a forest trust. A forest trust is a one-way or two-way transitive trust relationship between the forest root domains of two forests. The slide shows an example of a forest trust between the tailspintoys.com forest and the worldwideimporters.com forest.

A single forest trust relationship allows the authentication of a user in any domain by any other domain in either forest, assuming that the forest trust is two-way. If the forest trust is one-way, any user in any domain in the trusted forest can be authenticated by computers in the trusting forest. Forest trusts are significantly easier to establish, maintain, and administer than separate trust relationships between each of the domains in the forests. Forest trusts are particularly useful in scenarios involving cross-organization collaboration or mergers and acquisitions, or within a single organization that has more than one forest to isolate Active Directory data and services.

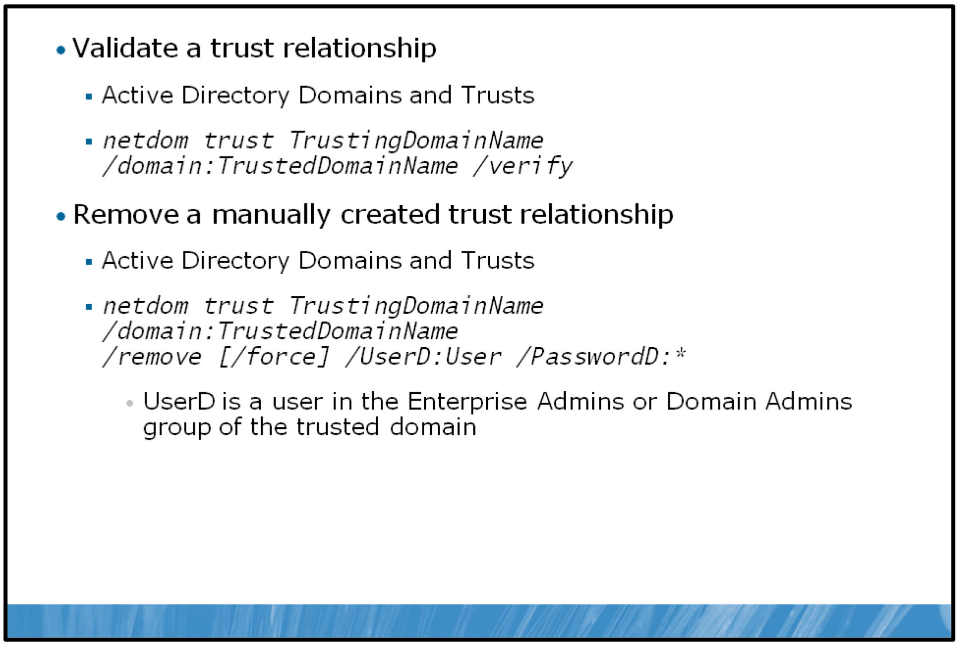
When you establish a forest trust relationship, domain quarantine, also called SID filtering, is enabled by default. Domain quarantine is discussed in the Domain Quarantine section.

You can specify whether the forest trust is one-way, incoming or outgoing, or two-way. As mentioned earlier, a forest trust is transitive, allowing all domains in a trusting forest to trust all domains in a trusted forest.

However, forest trusts are not themselves transitive. For example, if the tailspintoys.com forest trusts the worldwideimporters.com forest, and the worldwideimporters.com forest trusts the northwindtraders.com forest, those two trust relationships do not allow the tailspintoys.com forest to trust the northwindtraders.com forest. If you want those two forests to trust each other, you must create a specific forest trust between them.

Several requirements must be met before you can implement a forest trust. The forest functional level must be Windows Server 2003 or later. In addition, you must have a specific DNS infrastructure to support a forest trust.

Administer Trust Relationships



- Validate a trust relationship
  - Active Directory Domains and Trusts
  - *netdom trust TrustingDomainName /domain:TrustedDomainName /verify*
- Remove a manually created trust relationship
  - Active Directory Domains and Trusts
  - *netdom trust TrustingDomainName /domain:TrustedDomainName /remove [/force] /UserD:User /PasswordD:**
    - UserD is a user in the Enterprise Admins or Domain Admins group of the trusted domain

If you are concerned that a trust relationship is not functioning, you can validate a trust relationship between any two Windows domains. You cannot validate a trust relationship to a Kerberos v5 realm. To validate a trust relationship, complete the following steps:

1. Open Active Directory Domains and Trusts.
2. In the console tree, right-click the domain that contains the trust that you want to validate, and then click **Properties**.
3. Click the **Trusts** tab.
4. Select the trust you want to validate.
5. Click **Properties**.
6. Click **Validate**.
7. Do one of the following, and then click **OK**:
   - Click **Yes, Validate The Incoming Trust**. Enter credentials of the members of the Domain Admins or Enterprise Admins groups in the reciprocal domain.
   - Click **No, Do Not Validate The Incoming Trust**. Repeat this procedure for the reciprocal domain.

You can also verify a trust from the command prompt by typing the following command.

```
netdom trust TrustingDomainName /domain:TrustedDomainName /verify
```

There can also be reasons to remove a manually created trust. To do so, follow these steps:

1. Open Active Directory Domains and Trusts.

2. In the console tree, right-click the domain that contains the trust you want to validate, and then click **Properties**.
3. Click the **Trusts** tab.
4. Select the trust you want to remove.
5. Click **Remove**.
6. Do one of the following, and then click **OK**:
   - Click **Yes, Remove The Trust From Both The Local Domain And The Other Domain**. Enter the credentials of the members of the Domain Admins or Enterprise Admins groups in the reciprocal domain.
   - Click **No, Remove The Trust From The Local Domain Only**. Repeat this procedure for the reciprocal domain.

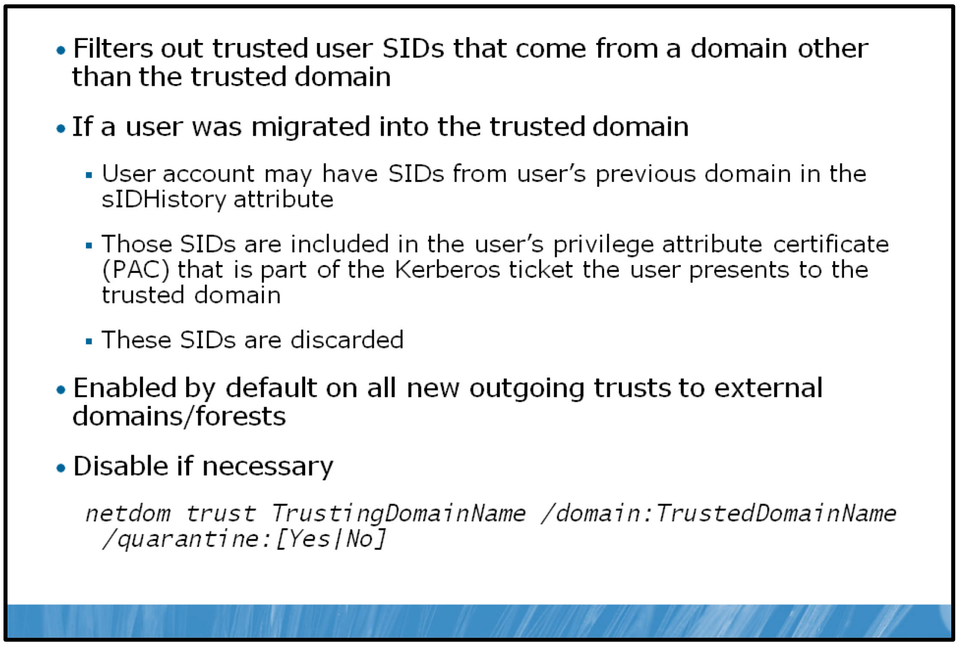
To delete a manually created trust from the command prompt, use the netdom.exe command with the following syntax.

```
netdom trust TrustingDomainName /domain:TrustedDomainName
/remove [/force] /UserD:User /PasswordD:*
```

The UserD parameter is a user with credentials in the Enterprise Admins or Domain Admins group of the trusted domain. Specifying the PasswordD:* parameter causes netdom.exe to prompt you for the password to the account. The /force switch is required when removing a realm trust.

Note The Windows Domain Manager, netdom.exe, and other command-line tools can be used to manage and test trust relationships. Visit http://go.microsoft.com/fwlink/?LinkId=168832 for details regarding these commands.

Domain Quarantine



- Filters out trusted user SIDs that come from a domain other than the trusted domain
- If a user was migrated into the trusted domain
  - User account may have SIDs from user's previous domain in the sIDHistory attribute
  - Those SIDs are included in the user's privilege attribute certificate (PAC) that is part of the Kerberos ticket the user presents to the trusted domain
  - These SIDs are discarded
- Enabled by default on all new outgoing trusts to external domains/forests
- Disable if necessary

```
netdom trust TrustingDomainName /domain:TrustedDomainName
 /quarantine:[Yes|No]
```

By default, domain quarantine, also called SID filtering, is enabled on all external and forest trusts. When a user is authenticated in a trusted domain, the user presents authorization data that includes the SIDs of the user's account in the groups to which the user belongs. Additionally, the user's authorization data includes security identifiers from other attributes of the user and his or her groups.

Some of the SIDs presented by the user from the trusted domain might not have been created in the trusted domain. For example, if a user is migrated from one domain into another, a new SID is assigned to the migrated account. The migrated account will, therefore, lose access to any resources that had permissions assigned to the SID of the user's former account. To enable the user to continue to access such resources, an administrator performing a migration can specify that the sIDHistory attribute of the user's migrated account will include the former account's SID. When the user attempts to connect to the resource, the original SID in the sIDHistory attribute will be authorized for access.

In a trusted domain scenario, it is possible that a rogue administrator could use administrative credentials in the trusted domain to load SIDs into the sIDHistory attribute of a user that are the same as SIDs of privileged accounts in your domain. That user would then have inappropriate levels of access to resources in your domain.

Domain quarantine prevents this by enabling the trusting domain to filter out SIDs from the trusted domain that are not the primary SIDs of security principals. Each SID includes the SID of the originating domain, so when a user from a trusted domain presents the list of the user's SIDs and the SIDs of the user's groups, SID filtering instructs the trusting domain to discard all SIDs without the domain SID of the trusted domain.

Domain quarantine is enabled by default for all outgoing trusts to external domains and forests. Disable domain quarantine only if one or more of the following are true:

- You have extremely high levels of confidence in the administrators of the trusted domain.
- Users or groups have been migrated to the trusted domain with their SID histories preserved, and you want to grant those users or groups permissions to resources in the trusting domain based on the sIDHistory attribute.

To disable domain quarantine, type the following command.

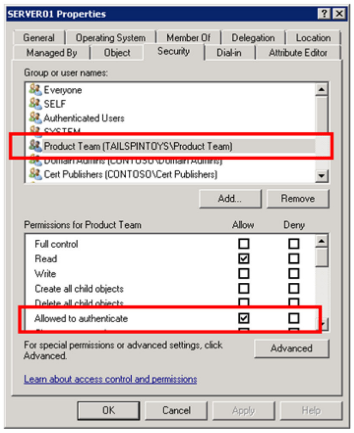
```
netdom trust TrustingDomainName /domain:TrustedDomainName /quarantine:no
```

To re-enable domain quarantine, type this command.

```
netdom trust TrustingDomainName /domain:TrustedDomainName /quarantine:yes
```

Resource Access for Users from Trusted Domains

- Giving trusted users access to resources
  - Authenticated Users
  - Add trusted identities to trusting domain's domain local groups
  - Add trusted identities to ACLs
- Selective authentication
  - Reduces the risk of exposure--for example, to Authenticated Users
  - You specify which trusted users are allowed to authenticate on a server-by-server (computer-by-computer) basis
  - Enables selective authentication in the properties of the trust
  - Gives users Allowed To Authenticate permission on the computer object in Active Directory



Group or user names:

Everyone
SELF
Authenticated Users
SYSTEM
Product Team (TAILSPINTOYS\Product Team)
Domain Admins (CONTOSO\Domain Admins)
Cert Publishers (CONTOSO\Cert Publishers)

Permissions for Product Team	Allow	Deny
Full control	□	□
Read	☑	□
Write	□	□
Create all child objects	□	□
Delete all child objects	□	□
Allowed to authenticate	☑	□

For special permissions or advanced settings, click Advanced.

Learn about access control and permissions

When you configure a trust relationship that enables your domain to trust another domain, you open up the possibility for users in the trusted domain to gain access to resources in your domain. The following sections examine components related to the security of a trusting domain's resources.

Authenticated Users

A trust relationship itself does not grant access to any resources; however, it is likely that by creating a trust relationship, users in the trusted domain will have immediate access to a number of your domain's resources. This is because many resources are secured with ACLs that give permissions to the Authenticated Users group.

Membership in Domain Local Groups

The best practice for managing access to a resource is to assign permissions to a domain local group. You can then nest users and groups from your domain into the domain local group and, thereby, grant them access to the resource. Domain local security groups can also include users and global groups from trusted domains as members. Therefore, the most manageable way to assign permissions to users in a trusted domain is to make them or their global groups members of a domain local group in your domain.

Add Trusted Identities to ACLs

You can also add users and global groups from a trusted domain directly to the ACLs of resources in a trusting domain. This approach is not as manageable as the previous method, using a domain local group, but it is possible.

Transitivity

When you create a realm trust, the trust is nontransitive by default. If you make it transitive, you open up the potential for users from domains and realms trusted by the Kerberos v5 realm to gain access to resources in your domain. Use nontransitive trusts unless you have a compelling business reason for a transitive realm trust.

Selective Authentication

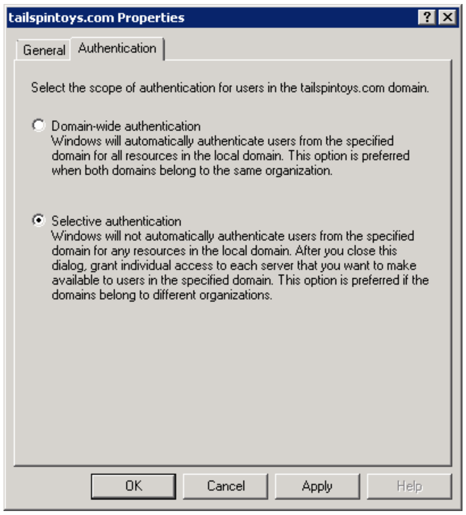
When you create an external trust or a forest trust, you can control the scope of authentication of trusted security principals. There are two modes of authentication for an external or forest trust:

- Selective authentication
- Domain-wide authentication (for an external trust) or forest-wide authentication (for a forest trust)

If you choose domain-wide or forest-wide authentication, all trusted users can be authenticated for access to services on all computers in the trusting domain. Trusted users can, therefore, be given permission to access resources anywhere in the trusting domain. With this authentication mode, you must have confidence in the security procedures of your enterprise and in the administrators who implement those procedures so that inappropriate access is not assigned to trusted users. Remember, for example, that users from a trusted domain or forest are considered Authenticated Users in the trusting domain, so any resource with permissions granted to Authenticated Users will be immediately accessible to trusted domain users if you choose domain-wide or forest-wide authentication.

If, however, you choose selective authentication, all users in the trusted domain are trusted identities; however, they are allowed to authenticate only for services on computers that you have specified. For example, imagine that you have an external trust with a partner organization's domain. You want to ensure that only users from the marketing group in the partner organization can access shared folders on only one of your many file servers. You can configure selective authentication for the trust relationship and then give the trusted users the right to authenticate only for that one file server.

To configure the authentication mode for a new outgoing trust, use the Outgoing Trust Authentication Level page of the New Trust Wizard. Configure the authentication level for an existing trust, open the properties of the trusting domain in Active Directory Domains and Trusts, select the trust relationship, click Properties, and then click the Authentication tab, shown in the illustration.



Select the scope of authentication for users in the tailspintoys.com domain.

○ Domain-wide authentication
Windows will automatically authenticate users from the specified domain for all resources in the local domain. This option is preferred when both domains belong to the same organization.

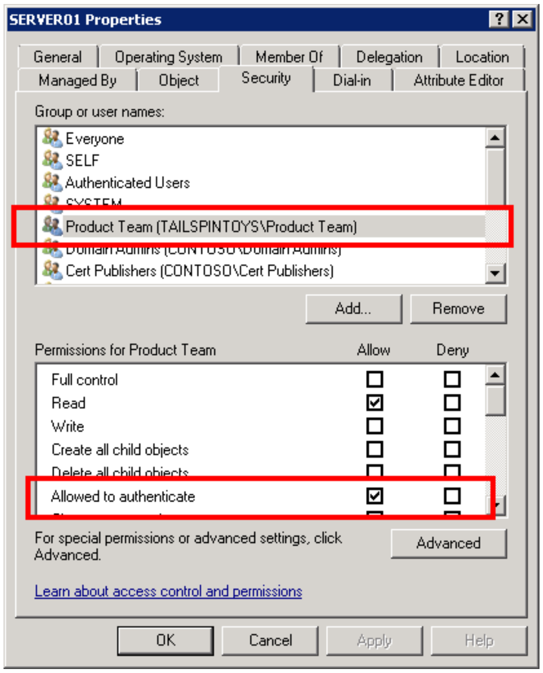
● Selective authentication
Windows will not automatically authenticate users from the specified domain for any resources in the local domain. After you close this dialog, grant individual access to each server that you want to make available to users in the specified domain. This option is preferred if the domains belong to different organizations.

After you have selected Selective authentication for the trust, no trusted users will be able to access resources in the trusting domain, even if those users have been given permissions.

The users must also be assigned the Allowed To Authenticate permission on the computer object in the domain.

To assign this permission:

1. Open the **Active Directory Users and Computers** snap-in and make sure that **Advanced Features** is selected on the **View** menu.
2. Open the properties of the computer to which trusted users should be allowed to authenticate—that is, the computer that trusted users will log on to or that contains resources to which trusted users have been given permissions.
3. On the **Security** tab, add the trusted users or a group that contains them and select the **Allow** check box for the **Allowed to authenticate** permission, as shown in the next illustration.



SERVER01 Properties

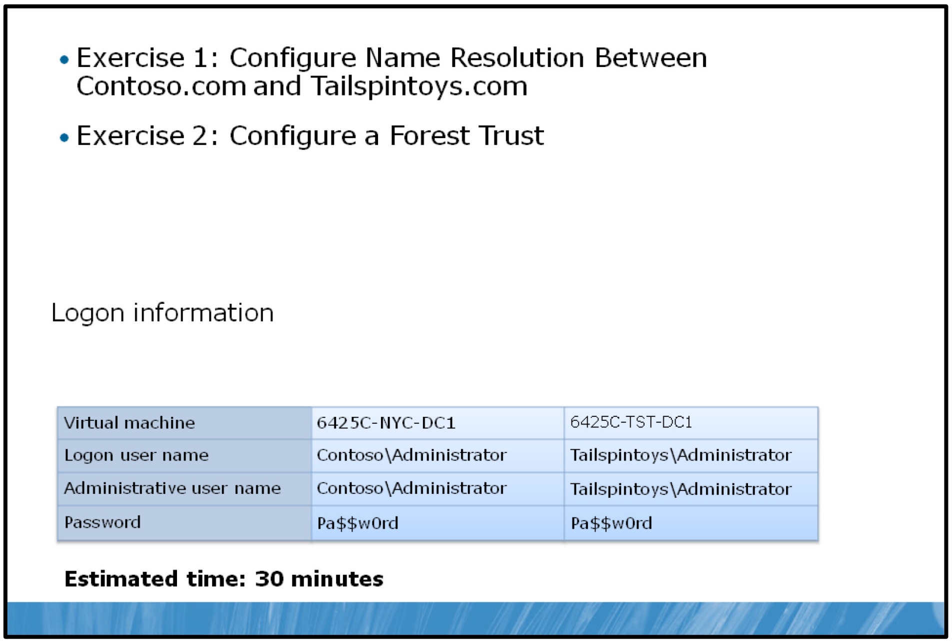
Group or user names:

Everyone
SELF
Authenticated Users
SYSTEM
Product Team (TAILSPINTOYS\Product Team)
Domain Admins (CONTOSO\Domain Admins)
Cert Publishers (CONTOSO\Cert Publishers)

Permissions for Product Team	Allow	Deny
Full control	□	□
Read	☑	□
Write	□	□
Create all child objects	□	□
Delete all child objects	□	□
Allowed to authenticate	☑	□

For special permissions or advanced settings, click Advanced.

Lab: Administer Trust Relationships



- Exercise 1: Configure Name Resolution Between Contoso.com and Tailspintoys.com
- Exercise 2: Configure a Forest Trust

Logon information

Virtual machine	6425C-NYC-DC1	6425C-TST-DC1
Logon user name	Contoso\Administrator	Tailspintoys\Administrator
Administrative user name	Contoso\Administrator	Tailspintoys\Administrator
Password	Pa$$w0rd	Pa$$w0rd

Estimated time: 30 minutes

Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. On the host computer, click **Start**, point to **Administrative Tools**, and then click **Hyper-V Manager**.
2. In Hyper-V™ Manager, click **6425C-NYC-DC1**, and in the Actions pane, click **Start**.
3. In the Actions pane, click **Connect**. Wait until the virtual machine starts.
4. Log on by using the following credentials:
   - User name: **Administrator**
   - Password: **Pa$$w0rd**
   - Domain: **Contoso**
5. Repeat steps 2 and3 for **6425C-TST-DC1.** Log on to **TST-DC1** as **Tailspintoys\Administrator,** with the password, **Pa$$w0rd.**

Lab Scenario

Contoso, Ltd has initiated a strategic partnership with
Tailspin Toys. Users from the two organizations will need to access files when collaborating on joint projects. You need to perform the following tasks:

- Configure name resolution between the two forests.
- Configure a forest trust relationship between Contoso.com and Tailspintoys.com.
- Configure Selective Authentication to only allow Tailspintoys.com domain users to access NYC-SVR1.

Exercise 1: Configure Name Resolution Between Contoso.com and Tailspintoys.com

In this exercise, you will configure conditional forwarding to provide name resolution between the Contoso.com domain and the Tailspintoys.com domain.

The main tasks for this exercise are as follows:

1. Configure DNS conditional forwarding on NYC-DC1.
2. Configure DNS conditional forwarding on TST-DC1.

▶ Task 1: Configure DNS conditional forwarding on NYC-DC1.

1. On NYC-DC1, open **DNS Manager**.
2. Configure a Conditional Forwarder with the following settings:
   - DNS Domain: **Tailspintoys.com**.
   - IP address of master servers: **10.0.0.31**

▶ Task 2: Configure DNS conditional forwarding on TST-DC1.

1. On TST-DC1, open **DNS Manager**.
2. Configure a Conditional Forwarder with the following settings:
   - DNS Domain: Contoso.com.
   - IP address of master servers: 10.0.0.10

Results: In this exercise, you configured name resolution between the Contoso.com domain and the Tailspintoys.com domain.

Exercise 2: Configure a Forest Trust

You need to configure a forest trust between Contoso.com and Tailspintoys.com.

The main tasks for this exercise are as follows:

1. Use the New Trust Wizard to create a forest trust.
2. Configure selective authentication.

▶ Task 1: Use the New Trust Wizard to create a forest trust.

1. On NYC-DC1, open the **Active Directory Domains and Trusts** console.
2. Start the **New Trust Wizard** and configure the following:
   - Trust Name: **Tailspintoys.com**
   - Trust Type: **Forest Trust**
   - Direction of Trust: **Two-way**
   - Sides of Trust: **Both this domain and the specified domain**
   - User Name: **Administrator**
   - Password: **Pa$$w0rd**
   - Outgoing Trust Authentication Level – Local Forest: **Forest-wide authentication**
   - Outgoing Trust Authentication Level – Specified Forest: **Forest-wide authentication**
   - Confirm both the outgoing and incoming trust

▶ Task 2: Configure selective authentication.

- On NYC-DC1, configure Selective Authentication to allow only Tailspintoys.com domain users to authenticate to NYC-SVR1, which is located in the Servers\File organizational unit.

Results: In this exercise, you created a forest trust and configured Selective Authentication.

▶ To revert the virtual machines

When you finish the lab, revert the virtual machines to their initial state. To do this, complete the following steps:

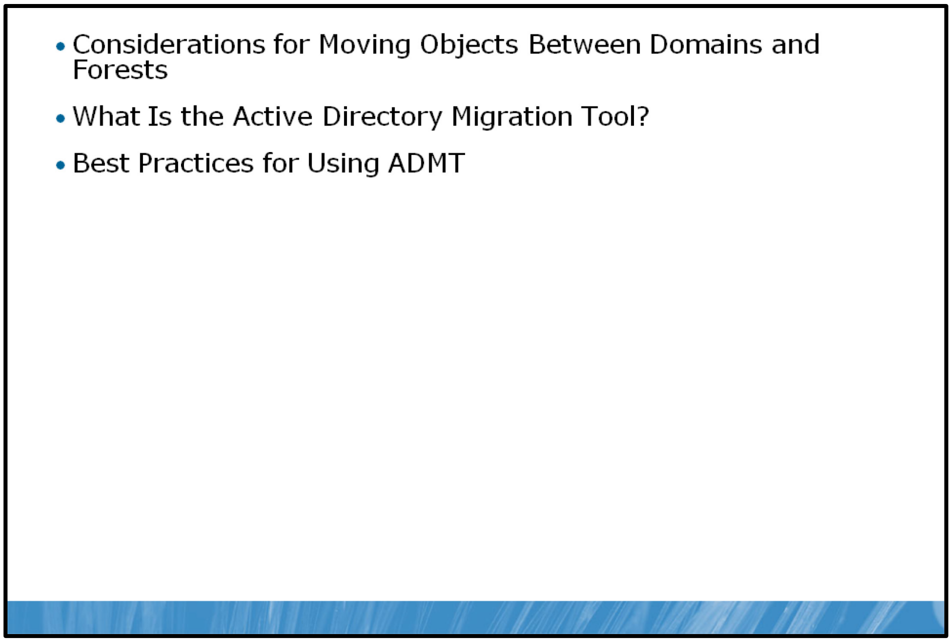
1. On the host computer, start Hyper-V Manager.
2. Right-click **6425C-NYC-DC1** in the **Virtual Machines** list, and then click **Revert**.
3. In the **Revert Virtual Machine** dialog box, click **Revert**.
4. Repeat these steps for **6425C-TST-DC1**.

Lab Review Questions

Question: How would you configure a forest trust with another organization if the organization does not provide you with their administrator credentials?

Question: What is the main benefit of Selective Authentication?

Lesson 3
Move Objects Between Domains and Forests



- Considerations for Moving Objects Between Domains and Forests
- What Is the Active Directory Migration Tool?
- Best Practices for Using ADMT

As your organization expands, you might need to restructure the current AD DS infrastructure. This can include migration of resources between AD DS domains in the same forest, or between domains located in different forests.

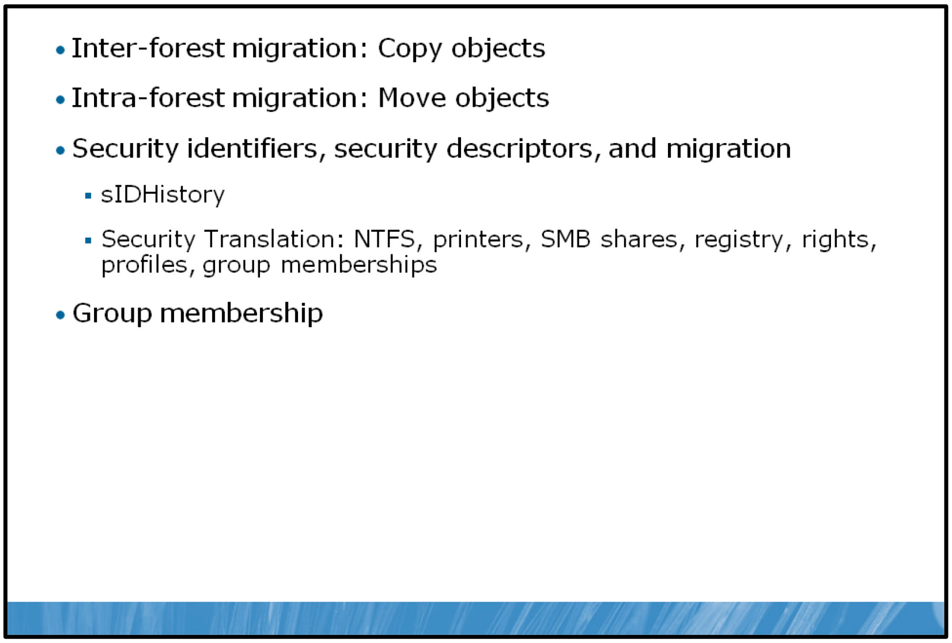
The Active Directory Migration Tool (ADMT) can be used to migrate objects between domains, and it provides the functionality to help ensure that users can maintain access to network resources throughout the migration process.

Objectives

After completing this lesson, you will be able to:

- Describe the considerations for moving objects between domains and forests.
- Describe the ADMT.
- Describe how ADMT works.
- Describe the best practices for using ADMT.

Considerations for Moving Objects Between Domains and Forests



- Inter-forest migration: Copy objects
- Intra-forest migration: Move objects
- Security identifiers, security descriptors, and migration
  - sIDHistory
  - Security Translation: NTFS, printers, SMB shares, registry, rights, profiles, group memberships
- Group membership

In multi-domain scenarios, you might need to move users, groups, or computers between domains or forests to support business operations. You might need to move large quantities of users, groups, or computers between domains or forests to implement mergers and acquisitions or to restructure your domain model.

In each of these tasks, you move or copy the accounts from one domain (the source domain) into another domain (the target domain). Domain restructuring terminology, concepts, and procedures apply to inter-forest migration between a Windows NT 4.0 or Active Directory source domain and an Active Directory target domain in a separate forest and to intra-forest migration—that is, the restructuring or moving of accounts between domains in the same forest.

An inter-forest domain restructure preserves the existing source domain and clones (or copies) accounts into the target domain. This nondestructive method enables an enterprise to time the transition and even migrate in phases. Operations go uninterrupted because both domains are maintained in parallel to support operations for users in either domain. This method also provides a level of rollback because the original environment remains unaltered in any significant way. After the migration is complete, you can simply decommission the source domain by moving any remaining accounts, member servers, and workstations into the new domain, and then taking source domain controllers offline, at which point you can redeploy those DCs for roles in the new domain.

An intra-forest migration involves moving objects from the source domain to the target domain without decommissioning the source domain. After you have migrated objects, you can restructure your domains to consolidate operations and build a domain and OU structure that more accurately reflects your administrative model. Many organizations consolidate multiple domains into one Active Directory domain. This consolidation can result in cost savings and simplified administration by reducing administrative complexity and the cost of supporting your Active Directory environment.

Security Identifiers and Migration

Uninterrupted resource access is the primary concern during any migration. Further, to perform a migration, you must be comfortable with the concepts of security identifiers (SIDs), tokens, ACLs, and sIDHistory.

SIDs are domain-unique values that are assigned to the accounts of security principals—users, groups, and computers, for example—when those accounts are created. When a user logs on, a token is generated that includes the primary SID of the user account and the SIDs of groups to which the user belongs. The token thus represents the user with all the SIDs associated with the user and the user's group memberships.

Resources are secured by using a security descriptor (SD) that describes the permissions, ownership, extended rights, and auditing of the resource. Within the SD are two ACLs. The system ACL (SACL) describes auditing. The discretionary ACL (DACL) describes resource access permissions. Many administrators and documents refer to the DACL as the ACL. The DACL lists permissions associated with security principals. Within the list, individual access control entries (ACEs) link a specific permission with the SID of a security principal. The ACE can be an Allow or Deny permission.

When a user attempts to access a resource, the Local Security Authority Subsystem (LSASS) compares the SIDs in the user's token with the SIDs in the ACEs in the resource's ACL.

When you migrate accounts to a new domain, the accounts are copied or cloned from the source domain to the target domain. New SIDs are generated for the accounts in the target domain, so the SIDs of new accounts will not be the same as the SIDs of the accounts in the source domain. Therefore, even though the cloned accounts have the same name and many of the same properties, because the SIDs are different, the accounts are technically different and will not have access to resources in the source domain. You have two ways to address this problem: *sIDHistory*or security translation.

sIDHistory

Enterprises typically prefer to take advantage of the sIDHistory attribute to perform effective domain restructuring. The capitalization, which appears odd, reflects the capitalization of the attribute in the Active Directory schema. An Active Directory security principal (which can be a user, group, or computer) has a principal SID and a sIDHistory attribute, which can contain one or more SIDs that are also associated with the account. When an account is copied to a target domain, the unique principal SID is generated by Active Directory in the target domain. Optionally, the sIDHistory attribute can be loaded with the SID of the account in the source domain. When a user logs on to an Active Directory domain, the user's token is populated with the principal SID and the sIDHistory of the user account and groups to which the user belongs. The LSASS uses the SIDs from the sIDHistory just like any other SID in the token to maintain the user's access to resources in the source domain.

Security Translation

Security translation is the process of examining each resource's SD, including its ACLs, identifying each SID that refers to an account in the source domain, and replacing that SID with the SID of the account in the target domain. The process of remapping ACLs (and other elements in the SD) to migrated accounts in the target domain is also called re-ACLing. Security translation or re-ACLing can be a tedious process to perform manually even in the simplest environment.

Migration tools such as the ADMT automate security translation. The ADMT can translate the SDs and policies of resources in the source domain to refer to the corresponding accounts in the target domain. Specifically, the ADMT can translate:

- File and folder permissions

- Printer permissions
- Share permissions
- Registry permissions
- User rights
- Local profiles, which involves changing file, folder, and registry permissions
- Group memberships

In most domain restructuring and migration projects, sIDHistory is used to maintain access and functionality during the migration. Then, security translation is performed.

Group Membership

The final concern related to resource access is that of group membership. Global groups can contain members only from the same domain. Therefore, if you clone a user to the target domain, the new user account cannot be a member of the global groups in the source domain to which the source user account belonged.

To address this issue in an inter-forest migration, first migrate global groups to the target domain. Those global groups will maintain the source groups' SIDs in their sIDHistory attributes, maintaining resource access. Then, migrate users. As you migrate users, the ADMT evaluates the membership of the source account and adds the new account to the same group in the target domain. If the group does not yet exist in the target domain, the ADMT can create it automatically. Ultimately, the user account in the target domain will belong to global groups in the target domain. The user and the user's groups will contain the SIDs of the source accounts in their sIDHistory attributes. Therefore, the user will be able to access resources in the source domain that have permissions assigned to the source accounts.

In an intra-forest migration, the process works differently. A global group is created in the target domain as a universal group so that it can contain users from both the source and the target domain. The new group gets a new SID, but its sIDHistory is populated with the SID of the global group in the source domain, thereby maintaining resource access for the new group. After all users have been migrated from the source to the target domain, the scope of the group is changed back to global.

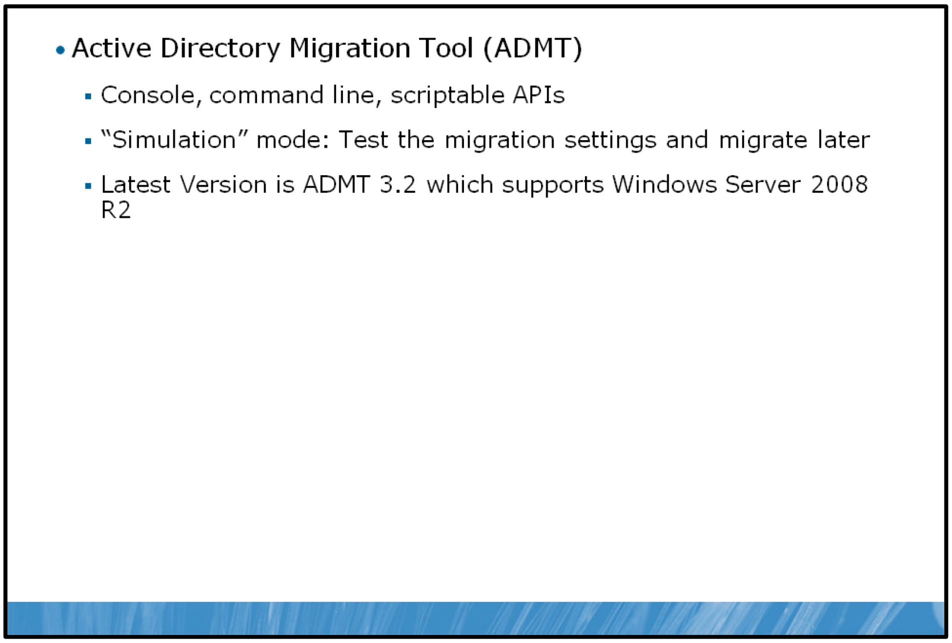
Other Migration Concerns

There are several issues that you must address in planning for and executing the migration of objects between domains and forests. Each of the concerns is detailed in the ADMT user guide, which is available from the ADMT download page listed earlier. Among the greatest concerns are:

- **Password migration.** The ADMT supports migrating user passwords; however, it cannot confirm that those passwords comply with the policies of the target domain regarding password length and complexity. Nonblank passwords will migrate regardless of the target domain password policy, and users will be able to log on with those passwords until they expire, at which time a new, compliant password must be created. If you are concerned about locking down the environment at the time of migration, this might not be a satisfactory process. You might instead want to let the ADMT configure complex passwords or script an initial password, and then force the user to change the password at the first logon.
- **Service accounts.** Services on domain computers might use domain-based user accounts for authentication. As those user accounts are migrated to the target domain, services must be updated with the new service account identity. The ADMT automates this process.

- **Objects that cannot be migrated.** Some objects cannot be seamlessly migrated. The ADMT cannot migrate built-in groups such as Domain Admins or the domain local Administrators group. The user guide provides details for working around this limitation.

What Is the Active Directory Migration Tool?



- Active Directory Migration Tool (ADMT)
  - Console, command line, scriptable APIs
  - "Simulation" mode: Test the migration settings and migrate later
  - Latest Version is ADMT 3.2 which supports Windows Server 2008 R2

The ADMT can perform object migration and security translation tasks. You can download the latest version from http://go.microsoft.com/fwlink/?LinkID=214206. On that page, you will also find a detailed guide to the tool.

You can use the ADMT to migrate objects between a source and a target domain. The migration can take place between domains in the same forest (an intra-forest migration) or between domains in different forests (an inter-forest migration). The ADMT provides wizards that automate migration tasks such as migrating users, groups, service accounts, computers, and trusts, and performing security translation. You can perform these tasks by using the ADMT console or the command line at which you can simplify and automate the admt.exe command with option files that specify parameters for the migration task. Then, using a simple text file, you can list objects to migrate rather than having to enter each object on the command line. The ADMT also provides interfaces that enable you to script migration tasks with languages such as Microsoft Visual Basic® Scripting Edition (VBScript). Run the ADMT console and open the online Help function for details about how to use the ADMT from the command line and about scripting the ADMT.

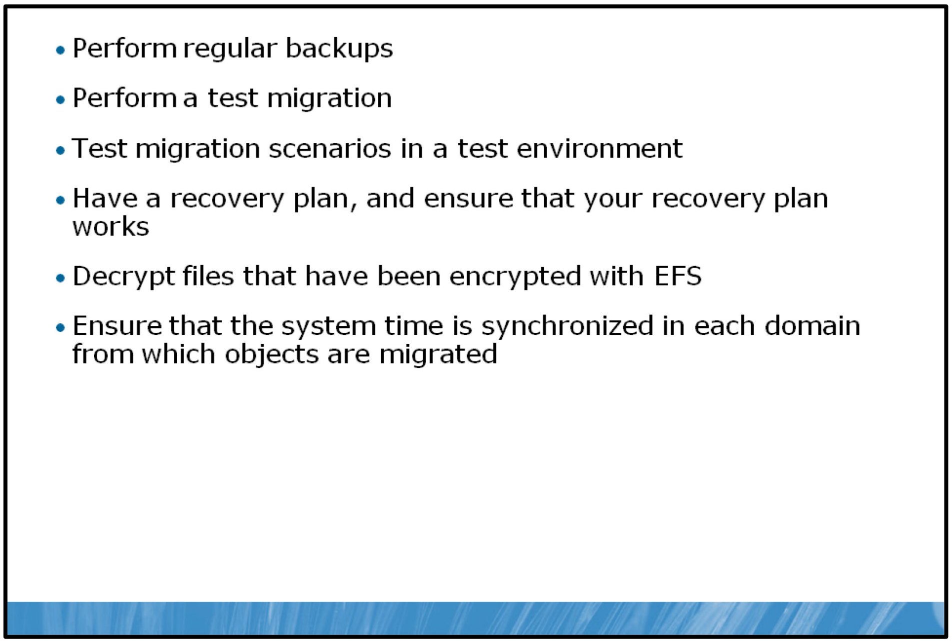
When you are performing migration tasks, the ADMT enables you to simulate the migration so that you can evaluate potential results and errors without making changes to the target domain. Wizards provide the Test the migration settings and migrate later option. You can then configure the migration task, test the settings, and review the log files and wizard-generated reports. After identifying and resolving any problems, you can perform the migration task. You will repeat this process of testing and analyzing results as you migrate users, groups, and computers and perform security translations.

ADMT 3.2 has the following requirements:

- ADMT can be installed on any computer capable of running the Windows Server 2008 R2 operating system, except RODCs or servers installed by using the Server Core installation option.

- The target domain must be running Windows Server 2003, Windows Server 2008, or Windows Server 2008 R2.
- The source domain must be running Windows Server 2003, Windows Server 2008, or Windows Server 2008 R2.
- The ADMT agent, installed by ADMT on computers in the source domains, can operate on computers running Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2.

Best Practices for Using ADMT



- Perform regular backups
- Perform a test migration
- Test migration scenarios in a test environment
- Have a recovery plan, and ensure that your recovery plan works
- Decrypt files that have been encrypted with EFS
- Ensure that the system time is synchronized in each domain from which objects are migrated

Before performing migration tasks by using ADMT, consider the following best practices:

- Perform regular backups of domain controllers in both the source and target domains throughout the course of the migrations. If you are migrating computers that contain file shares to perform security translation, you should also back up those computers throughout migration.
- Before you begin a migration, perform a test migration by creating a test user, adding the test user to the appropriate global groups, and then verifying resource access before and after migration.
- Test your migration scenarios in a test environment before migrating objects in the production environment.
- Have a recovery plan, and ensure that your recovery plan works during the test phase of your migration.
- Decrypt files that have been encrypted by means of Encrypting File System (EFS). Failure to decrypt encrypted files will result in loss of access to encrypted files after migration. Be sure to communicate to end users that they must decrypt any encrypted files or they will lose access to those files.
- Ensure that the system time is synchronized in each domain from which objects are migrated. Kerberos authentication fails if time is skewed.