# Introduction

## Background

**Active Directory** (**AD**) is a directory service that Microsoft developed for Windows domain networks. It is included in most Windows Server operating systems as a set of processes and services. Initially, Active Directory was only in charge of centralized domain management. Starting with Windows Server 2008, however, Active Directory became an umbrella title for a broad range of directory-based identity-related services.

A server running Active Directory Domain Services (AD DS) is called a domain controller. It authenticates and authorizes all users and computers in a Windows domain type network—assigning and enforcing security policies for all user and computers objects. Also, it allows management and storage of information, provides authentication and authorization mechanisms, and establishes a framework to deploy other related services: Certificate Services, Active Directory Federation Services, Lightweight Directory Services and Rights Management Services.

## History

Active Directory, like many information-technology efforts, originated out of a democratization of design using Request for Comments or RFCs. The Internet Engineering Task Force (IETF), which oversees the RFC process, has accepted numerous RFCs initiated by widespread participants. Active Directory incorporates decades of communication technologies into the overarching Active Directory concept then makes improvements upon them. For example, LDAP underpins Active Directory. Also X.500 directories and the Organizational Unit preceded the Active Directory concept that makes use of those methods. The LDAP concept began to emerge even before the founding of Microsoft in April 1975, with RFCs as early as 1971. RFCs contributing to LDAP include RFC 1823 (on the LDAP API, August 1995), RFC 2307, RFC 3062, and RFC 4533.

Microsoft previewed Active Directory in 1999, released it first with Windows 2000 Server edition, and revised it to extend functionality and improve administration in Windows Server 2003. Additional improvements came with subsequent versions of Windows Server. In Windows Server 2008, additional services were added to Active Directory, such as Active Directory Federation Services. The part of the directory in charge of management of domains, which was previously a core part of the operating system, was renamed Active Directory Domain Services (ADDS) and became a server role like others. "Active Directory" became the umbrella title of a broader range of directory-based services.

## Active Directory Services

Active Directory Services consist of multiple directory services. The best known is Active Directory Domain Services, commonly abbreviated as AD DS or simply AD.

### Domain Services

Active Directory Domain Services (AD DS) is the cornerstone of every Windows domain network. It stores information about members of the domain, including devices and users, verifies their credentials and defines their access rights. The server running this service is called a domain controller. A domain controller is contacted when a user logs into a device, accesses another device across the network, or runs a line-of-business Metro-style app sideloaded into a device.

Other Active Directory services (excluding LDS, as described below) as well as most of Microsoft server technologies rely on or use Domain Services; examples include Group Policy, Encrypting File System, BitLocker, Domain Name Services, Remote Desktop Services, Exchange Server and SharePoint Server.

### Lightweight Directory Services

Active Directory Lightweight Directory Services (AD LDS), formerly known as Active Directory Application Mode (ADAM), is a light-weight implementation of AD DS. AD LDS runs as a service on Windows Server. AD LDS shares the code base with AD DS and provides the same functionality, including an identical API, but does not require the creation of domains or domain controllers. It provides a Data Store for storage of directory data and a Directory Service with an LDAP Directory Service Interface. Unlike AD DS, however, multiple AD LDS instances can run on the same server.

### Certificate Services

Active Directory Certificate Services (AD CS) establishes an on-premises public key infrastructure. It can create, validate and revoke public key certificates for internal uses of an organization. These certificates can be used to encrypt files (when used with Encrypting File System), emails (per S/MIME standard), and network traffic (when used by virtual private networks, Transport Layer Security protocol or IPSec protocol).

AD CS predates Windows Server 2008, but its name was simply Certificate Services.

AD CS requires an AD DS infrastructure.

### Federation Services

Active Directory Federation Services (AD FS) is a single sign-on service. With an AD FS infrastructure in place, users may use several web-based services (e.g. internet forum, blog, online shopping, webmail) or network resources using only one set of credentials stored at a central location, as opposed to having to be granted a dedicated set of credentials for each service. AD FS's purpose is an extension of that of AD DS: The latter enables users to authenticate with and use the devices that are part of the same network, using one set of credentials. The former enables them to use the same set of credentials in a different network.

As the name suggests, AD FS works based on the concept of federated identity.

AD FS requires an AD DS infrastructure, although its federation partner may not.

### Rights Management Services

Active Directory Rights Management Services (AD RMS, known as Rights Management Services or RMS before Windows Server 2008) is a server software for information rights management shipped with Windows Server. It uses encryption and a form of selective functionality denial for limiting access to documents such as corporate e-mails, Microsoft Word documents, and web pages, and the operations authorized users can perform on them.

## Logical Structure

As a directory service, an Active Directory instance consists of a database and corresponding executable code responsible for servicing requests and maintaining the database. The executable part, known as Directory System Agent, is a collection of Windows services and processes that run on Windows 2000 and later. Objects in Active Directory databases can be accessed via LDAP, ADSI (a component object model interface), messaging API and Security Accounts Manager services.

### Objects

Active Directory structures are arrangements of information about objects. The objects fall into two broad categories: resources (e.g., printers) and security principals (user or computer accounts and groups). Security principals are assigned unique security identifiers (SIDs).

Each object represents a single entity—whether a user, a computer, a printer, or a group—and its attributes. Certain objects can contain other objects. An object is uniquely identified by its name and has a set of attributes—the characteristics and information that the object represents— defined by a schema, which also determines the kinds of objects that can be stored in Active Directory.

The schema object lets administrators extend or modify the schema when necessary. However, because each schema object is integral to the definition of Active Directory objects, deactivating or changing these objects can fundamentally change or disrupt a deployment. Schema changes automatically propagate throughout the system. Once created, an object can only be deactivated—not deleted. Changing the schema usually requires planning.

### Forests, trees and domains

The Active Directory framework that holds the objects can be viewed at a number of levels. The forest, tree, and domain are the logical divisions in an Active Directory network.

Within a deployment, objects are grouped into domains. The objects for a single domain are stored in a single database (which can be replicated). Domains are identified by their DNS name structure, the namespace.

A domain is defined as a logical group of network objects (computers, users, devices) that share the same Active Directory database.

A tree is a collection of one or more domains and domain trees in a contiguous namespace, and is linked in a transitive trust hierarchy.

At the top of the structure is the forest. A forest is a collection of trees that share a common global catalog, directory schema, logical structure, and directory configuration. The forest represents the security boundary within which users, computers, groups, and other objects are accessible.

### Organizational units

The objects held within a domain can be grouped into Organizational Units (OUs). OUs can provide hierarchy to a domain, ease its administration, and can resemble the organization's structure in managerial or geographical terms. OUs can contain other OUs—domains are containers in this sense. Microsoft recommends using OUs rather than domains for structure and to simplify the implementation of policies and administration. The OU is the recommended level at which to apply group policies, which are Active Directory objects formally named Group Policy Objects (GPOs), although policies can also be applied to domains or sites (see below). The OU is the level at which administrative powers are commonly delegated, but delegation can be performed on individual objects or attributes as well.

Organizational units do not each have a separate namespace; e.g. user accounts with an identical username (sAMAccountName) in separate OUs within a domain are not allowed, such as "fred.staff-ou.domain" and "fred.student-ou.domain", where "staff-ou" and "student-ou" are the OUs. This is because sAMAccountName, a user object attribute, must be unique within the domain. However, two users in different OUs can have the same Common Name (CN), the name under which they are stored in the directory itself.

In general, the reason for this lack of allowance for duplicate names through hierarchical directory placement, is that Microsoft primarily relies on the principles of NetBIOS, which is a flat-file method of network object management that for Microsoft software, goes all the way back to Windows NT 3.1 and MS-DOS LAN Manager. Allowing for duplication of object names in the directory, or completely removing the use of NetBIOS names, would prevent backward compatibility with legacy software and equipment.

As the number of users in a domain increases, conventions such as "first initial, middle initial, last name" (Western order) or the reverse (Eastern order) fail for common family names like Li (李), Smith or Garcia. Workarounds include adding a digit to the end of the username. Alternatives include creating a separate ID system of unique employee/student id numbers to use as account names in place of actual user's names, and allowing users to nominate their preferred word sequence within an acceptable use policy. These workarounds have caused issues with more complex systems. As on premise and cloud systems have become more interconnected and complex, a need for the SamAccountName, Email ID, User Principal Name, and SIP ID to match has firmly established. Systems where this is not the case typically have increased user difficulty and administrative overhead.

Because duplicate usernames cannot exist within a domain and with increased system complexity, account name generation poses a significant challenge for large organizations that cannot be easily subdivided into separate domains, such as students in a public school system or university who must be able to use any computer across the network.

#### Shadow groups

In Active Directory, organizational units (OUs) cannot be assigned as owners or trustees. Only groups are selectable, and members of OUs cannot be collectively assigned rights to directory objects.

In Microsoft's Active Directory, OUs do not confer access permissions, and objects placed within OUs are not automatically assigned access privileges based on their containing OU. This is a design limitation specific to Active Directory.

Active Directory requires a separate step for an administrator to assign an object in an OU as a member of a group also within that OU. Relying on OU location alone to determine access permissions is unreliable, because the object may not have been assigned to the group object for that OU.

A common workaround for an Active Directory administrator is to write a custom PowerShell or Visual Basic script to automatically create and maintain a user group for each OU in their directory. The scripts are run periodically to update the group to match the OU's account membership, but are unable to instantly update the security groups anytime the directory changes, as occurs in competing directories where security is directly implemented into the directory itself. Such groups are known as Shadow Groups. Once created, these shadow groups are selectable in place of the OU in the administrative tools.

The following PowerShell script can create such shadow groups:

1. Import-Module ActiveDirectory
2. $groupname = PseudoDynamicGroup
3. $users = Get-ADUser -Filter \* -SearchBase "ou=desiredUsers,dc=domain,dc=tld"
4. foreach($user in $users)
5. {
6. Add-ADGroupMember -Identity $groupname -Member $user.samaccountname -ErrorAction SilentlyContinue
7. }
8. $members = Get-ADGroupMember -Identity $groupname
9. foreach($member in $members)
10. {
11. if($member.distinguishedname -notlike "\*ou=desiredUsers,dc=domain,dc=tld\*")
12. {
13. Remove-ADGroupMember -Identity $groupname -Member $member.samaccountname
14. }
15. }

Microsoft refers to shadow groups in the Server 2008 Reference documentation, but does not explain how to create them. There are no built-in server methods or console snap-ins for managing shadow groups.

The division of an organization's information infrastructure into a hierarchy of one or more domains and top-level OUs is a key decision. Common models are by business unit, by geographical location, by IT Service, or by object type and hybrids of these. OUs should be structured primarily to facilitate administrative delegation, and secondarily, to facilitate group policy application. Although OUs form an administrative boundary, the only true security boundary is the forest itself and an administrator of any domain in the forest must be trusted across all domains in the forest.

### Partitions

The Active Directory database is organized in partitions, each holding specific object types and following a specific replication pattern. Microsoft often refers to these partitions as 'naming contexts'. The 'Schema' partition contains the definition of object classes and attributes within the Forest. The 'Configuration' partition contains information on the physical structure and configuration of the forest (such as the site topology). Both replicate to all domains in the Forest. The 'Domain' partition holds all objects created in that domain and replicates only within its domain.

## Physical Structure

Sites are physical (rather than logical) groupings defined by one or more IP subnets. AD also holds the definitions of connections, distinguishing low-speed (e.g., WAN, VPN) from high-speed (e.g., LAN) links. Site definitions are independent of the domain and OU structure and are common across the forest. Sites are used to control network traffic generated by replication and also to refer clients to the nearest domain controllers (DCs). Microsoft Exchange Server 2007 and higher uses the site topology for mail routing. Policies can also be defined at the site level.

Physically, the Active Directory information is held on one or more peer domain controllers, replacing the NT PDC/BDC model. Each DC has a copy of the Active Directory. Servers joined to Active Directory that are not domain controllers are called Member Servers. A subset of objects in the domain partition replicate to domain controllers that are configured as global catalogs. Global catalog (GC) servers provide a global listing of all objects in the Forest. Global Catalog servers replicate to themselves all objects from all domains and hence, provide a global listing of objects in the forest. However, to minimize replication traffic and keep the GC's database small, only selected attributes of each object are replicated. This is called the partial attribute set (PAS). The PAS can be modified by modifying the schema and marking attributes for replication to the GC. Earlier versions of Windows used NetBIOS to communicate. Active Directory is fully integrated with DNS and requires TCP/IP—DNS. To be fully functional, the DNS server must support SRV resource records, also known as service records.

#### Replication

Active Directory synchronizes changes using multi-master replication. Replication by default is 'pull' rather than 'push', meaning that replicas pull changes from the server where the change was effected. The Knowledge Consistency Checker (KCC) creates a replication topology of site links using the defined sites to manage traffic. Intrasite replication is frequent and automatic as a result of change notification, which triggers peers to begin a pull replication cycle. Intersite replication intervals are typically less frequent and do not use change notification by default, although this is configurable and can be made identical to intrasite replication.

Each link can have a 'cost' (e.g., DS3, T1, ISDN etc.) and the KCC alters the site link topology accordingly. Replication may occur transitively through several site links on same-protocol site link bridges, if the cost is low, although KCC automatically costs a direct site-to-site link lower than transitive connections. Site-to-site replication can be configured to occur between a bridgehead server in each site, which then replicates the changes to other DCs within the site. Replication for Active Directory zones is automatically configured when DNS is activated in the domain based by site.

Replication of Active Directory uses Remote Procedure Calls (RPC) over IP (RPC/IP). Between Sites SMTP can be used for replication, but only for changes in the Schema, Configuration, or Partial Attribute Set (Global Catalog) GCs. SMTP cannot be used for replicating the default Domain partition.

## Database

The Active-Directory database, the directory store, in Windows 2000 Server uses the JET Blue-based Extensible Storage Engine (ESE98) and is limited to 16 terabytes and 2 billion objects (but only 1 billion security principals) in each domain controller's database. Microsoft has created NTDS databases with more than 2 billion objects. (NT4's Security Account Manager could support no more than 40,000 objects). Called NTDS.DIT, it has two main tables: the data table and the link table. Windows Server 2003 added a third main table for security descriptor single instancing.

## Single Server Operations

Flexible Single Master Operations Roles (FSMO, pronounced "fizz-mo") operations are also known as operations master roles. Although domain controllers allow simultaneous updates in multiple places, certain operations are supported only on a single server. These operations are performed using the roles listed below:

|  |  |  |
| --- | --- | --- |
| Role Name | Scope | Description |
| Schema Master | 1 per forest | Schema modifications |
| Domain Naming Master | 1 per forest | Addition and removal of domains if present in root domain |
| PDC Emulator | 1 per domain | Provides backwards compatibility for NT4 clients for PDC operations (like password changes). The PDC runs domain specific processes such as the Security Descriptor Propagator (SDP), and is the master time server within the domain. It also handles external trusts, the DFS consistency check, holds current passwords and manages all GPOs as default server. |
| RID Master | 1 per domain | Allocates pools of unique identifiers to domain controllers for use when creating objects |
| Infrastructure Master | 1 per domain/partition | Synchronizes cross-domain group membership changes. The infrastructure master should not be run on a global catalog server (GCS) unless all DCs are also GCs, or the environment consists of a single domain.  The Infrastructure Master role as described above is only for the domain partition (default naming context), netdom query fsmo and ntdsutil will only query the domain partition. However, every application partition, including Forest and Domain-level DNS domain zones has its own Infrastructure Master. The holder of this role is stored in the fSMORoleOwner attribute of the Infrastructure object in the root of the partition, it can be modified with ADSIEdit, for example one can modify the fSMORoleOwner attribute of the CN=Infrastructure,DC=DomainDnsZones,DC=yourdomain,DC=tld object to CN=NTDSSettings,CN=Name\_of\_DC,CN=Servers,CN=DRSite,CN=Sites,CN=Configuration,DC=Yourdomain,DC=TLD. |

## Trusting

To allow users in one domain to access resources in another, Active Directory uses trusts.

Trusts inside a forest are automatically created when domains are created. The forest sets the default boundaries of trust, and implicit, transitive trust is automatic for all domains within a forest.

**Terminology**

**One-way trust**

One domain allows access to users on another domain, but the other domain does not allow access to users on the first domain.

**Two-way trust**

Two domains allow access to users on both domains.

**Trusted domain**

The domain that is trusted; whose users have access to the trusting domain.

**Transitive trust**

A trust that can extend beyond two domains to other trusted domains in the forest.

**Intransitive trust**

A one way trust that does not extend beyond two domains.

**Explicit trust**

A trust that an admin creates. It is not transitive and is one way only.

**Cross-link trust**

An explicit trust between domains in different trees or in the same tree when a descendant/ancestor (child/parent) relationship does not exist between the two domains.

**Shortcut**

Joins two domains in different trees, transitive, one- or two-way.

**Forest trust**

Applies to the entire forest. Transitive, one- or two-way.

**Realm**

Can be transitive or nontransitive (intransitive), one- or two-way.

**External**

Connect to other forests or non-AD domains. Nontransitive, one- or two-way.

**PAM trust**

A one-way trust used by Microsoft Identity Manager from a (possibly low-level) production forest to a (Windows Server 2016 functionality level) 'bastion' forest, which issues time-limited group memberships.

**Forest trusts**

Windows Server 2003 introduced the forest root trust. This trust can be used to connect Windows Server 2003 forests if they are operating at the 2003 forest functional level. Authentication across this type of trust is Kerberos-based (as opposed to NTLM).

Forest trusts are transitive for all the domains the trusted forests. However, forest trusts are not transitive between forests.

Example: Suppose that a two-way transitive forest trust exists between the forest root domains in Forest A and Forest B, and another two-way transitive forest trust exists between the forest root domains in Forest B and Forest C. Such a configuration lets users in Forest B access resources in any domain in either Forest A or Forest C, and users in Forest A or C can access resources in any domain in Forest B. However, it does not let users in Forest A access resources in Forest C, or vice versa. To let users in Forest A and Forest C share resources, a two-way transitive trust must exist between both forests.

# Overview