CHAPTER 1

Understanding Active Directory
This chapter provides an introduction to Active Directory, discussing:

- Benefits of using Active Directory services
- Windows 2000 and Active Directory features
- X.500 conventions and attributes
- Lightweight Directory Access Protocol (LDAP)
- Microsoft Management Console (MMC)

**In the Beginning:**

**Before Directory Services**

Remember when you had one server and a few clients, and in hindsight, network administration was easy? Then the server’s hard drives became full, so a second server was added, and each user had to remember two passwords and login names. Over time with many servers, the end user and support staff became unhappy with the complicated network they had to use, administer, and share.

In 1993 Microsoft released NT with support for a single logon point using multiple servers employing *domain architecture* for a shared security database between multiple servers as shown in Figure 1.1.

**Figure 1.1**

The multiple master domain was common in NT design.
Understanding Active Directory

In the subsequent NT 3.51 and 4.0 versions, the security system was (and still is) stored and maintained on a domain controller called a primary domain controller, and replicated in read-only format to any available backup domain controllers.

Once any primary or backup domain controller within the users' defined domain authenticated the known user with a valid user account, that user could logon to the network. Another advantage that Microsoft enjoyed was perfect timing; the emergence of NT 3.51/4.0 was its integration with TCP/IP as the use of the Internet began to explode. One-way trusts were then used to link separate domains, creating potentially large network environments where the rules could and did change from domain to domain.

The major problem with the available domain structures supported by NT was that security was applied only at the domain level, making it impossible to delegate security effectively to other sub-administrators for control of the security principals user and group accounts, and computer accounts. The end result was a large sprawling network topology with multiple master domains with many one-way trusts left to manage and maintain; in many cases it just wasn't working well, especially in the network environments linked across Wide Area Network (WAN) links.

Novell NetWare and NDS

Novell was the first vendor with a large network market share to offer an X.500 directory service. Currently there are over 70 million NDS users worldwide. NDS can also run on any combination of NetWare, Windows NT, Linux, Solaris, IBM’s AIX, and OS/390 and can be accessed from DOS, Windows, UNIX, Linux, and Macintosh systems. As a result, Microsoft began to feel threatened, since NT was not as scalable as NDS. In 1995, the availability of Novell NetWare Version 4.0 introduced the network directory service (NDS), though it had no initial native support for TCP/IP. The other problem area for Novell was marketing NDS properly; NetWare had a very important directory service structure that would eventually be copied by Microsoft, but Novell couldn’t seem to educate the masses about the benefits of using NDS.

Instead, a majority of NetWare 3.X network administrators and companies began drifting toward Microsoft’s applications, browsers, and NOS. With the current version of NetWare 5.1, TCP/IP, rather than IPX can now be the native protocol used, and the many new NetWare solutions
appearing are based on the initial directory service model: DirXML Metadirectory, eDirectory, and NDS Corporate Edition.

However Windows NT 4.0 now controls the small and mid-size network market that NetWare had locked up for years. And Active Directory will probably be the final nail in the coffin for NetWare’s dominance of the mid-size networking market as a one-stop solution provider.

Active Directory Structure

The basic design and layout of Active Directory is based on the X.500 schema (think of a blueprint, or a building code as an analogy for the schema) just as NDS is. Microsoft adds many additional object classes, objects, and attributes into the schema once Exchange 2000 is deployed on top of Windows 2000; in fact, the size of the schema is more than doubled, and this is for just one application! Active Directory services are scalable and can be deployed in both mid-size and large enterprise network environments; however, the smaller network with just a few servers will also benefit, especially if Exchange is the preferred mail server.

Active Directory Provides Central Control

Using a directory service allows your organization’s administrators to manage all supported network resources across the entire directory structure. Through a single point of access, the administration of a worldwide Windows 2000 network is very doable. It all depends on your company’s current administrative practices and preferences. Certainly central control is a huge administrative benefit. And with Active Directory, security can be enforced across the entire directory structure, protecting resources, network services, and data files from the outside public world and also from users on your network by many granular security levels from one location. However, there are thousands of security settings that can be deployed, from the site level down to the single object and attribute level, so the setting of security is a huge job. At least we now have the ability to handle security properly, and we couldn’t say that with NT 4.0. Active Directory can also be distributed across a large network and WAN hierarchy, incorporating forests and trees of domains at multiple sites; these concepts will be discussed fully in the chapters to follow.
Improved Fault Tolerance

Multimaster directory replication between multiple domains supported by multiple domain controllers allows a much higher level of fault tolerance, and improved network traffic flow we could only dream about back in the NT 4.0 world. In Windows 2000, Domain Controller roles are now combined into one model, the Domain Controller (DC), since Primary Domain Controllers (PDC) and Backup Domain Controllers (BDC) are not used in native Windows 2000 networks. All Windows 2000 servers with Active Directory installed are Domain Controllers. However, Windows 2000 Domain Controllers can emulate the PDC role for the support of NT 4.0 BDCs, allowing replication of security information in the default mixed mode to a down-level NT 4.0 network; this permits Windows 2000 servers and NT 4.0 servers to co-exist during the migration process.

Industry Support

Active Directory also has ever-growing industry-wide support. Microsoft has relationships with many hundreds of hardware vendors and software developers who have bought into the Active Directory concept. This level of third-party support is something that NetWare never managed to achieve; however, Microsoft has developed into both an application developer and a network operating system provider that is proving impossible to stop. The key relationship to watch is Microsoft and Cisco. They are currently in a hush-hush relationship, and some of the most important security features of Windows 2000 (Quality of Service [QoS] and IPSec) were integrated and designed with Cisco’s help. Think of it this way: Microsoft wants to be your network plumbing. What better way to accomplish this task than to partner with the existing network plumber, Cisco? Once Windows 2000 is inside routers, it can then control your telephone system, alarm system, in fact any integrated system. Read the white papers on www.microsoft.net; you may be surprised. The takeover of the desktop and Internet browser will seem like child’s play a few years from now. This is a 20-year plan that’s just starting, and the first step is entrenching AD as the corporate network structure.
Application Software and Active Directory

Many versions of Microsoft’s popular application software will first support, then integrate with, and eventually require, Active Directory and Windows 2000 fully to deploy and provide all of their features. Exchange Server 2000, SQL Server, Proxy Server, and Systems Management Server 3.0 (Code named Emerald) are but the first wave.

- **SQL Server**—In future releases of SQL, server security will integrate completely with Active Directory.
- **Proxy Server**—Future versions of Proxy Server integrated with Active Directory security will be a big plus since the replicated active directory database combined with Kerberos security will provide a system-wide security level.
- **Systems Management Server 3.0**—It’s hard to imagine just what SMS 3.0 can offer—software deployment, remote access, and software licensing are all part of Windows 2000. Inventory control is not, however.

Exchange Server 2000 will be backward compatible with Exchange 5.5. Microsoft offers a directory migration tool to migrate the contents of the Exchange folder to the Active Directory folder.

Active Directory Features

Group Policy

The biggest feature of AD is group policy, a bizarre name, considering it is deployed for users and computer systems. Group policy (GP) works on many levels of control, from the site level, which may be your entire company, down to the domain level, then at the organizational unit level and, if desired, also at the local workstation level. You may be familiar with system policies that were available with Windows NT 4.0; these settings, plus a lot more, can be controlled and managed with group policy.
Understanding Active Directory

GP can be installed on a local Windows 2000 Professional node, but it is much more powerful when it is deployed through Active Directory as shown in Figure 1.2.

As mentioned, group policy can be deployed at several administrative levels: starting at sites, then in domains, and finally in organizational units. This order of deployment through inheritance is very important. Group policy can be deployed to take advantage of the areas of Windows 2000 shown in Table 1.1.

**TABLE 1.1**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ADM</strong></td>
<td>Administrative templates similar to NT 4.0 system policy settings but with enhancements</td>
</tr>
<tr>
<td><strong>Applications</strong></td>
<td>With the Windows Installer advertisements, applications and operating system updates can be performed automatically</td>
</tr>
<tr>
<td><strong>Files</strong></td>
<td>Files can be deployed to the client at logon</td>
</tr>
<tr>
<td><strong>Machine</strong></td>
<td>Contains settings and changes for the computer</td>
</tr>
<tr>
<td><strong>Scripts</strong></td>
<td>Startup and shutdown scripts can be used for computer configuration. Logon and logoff scripts can be use for the end user; they can be written in VBScript, Jscript, batch files, and WSH (Windows Scripting Host)</td>
</tr>
<tr>
<td><strong>User</strong></td>
<td>Contains settings and changes for the user and group of users</td>
</tr>
</tbody>
</table>
There are over 1,000 possible settings that can be deployed through group policy for the Windows 2000 operating system alone.

Folder Redirection

To a network client, a folder may appear to be locally stored on a Windows 2000 Professional client, when in fact it could be located on a network server as detailed in Figure 1.3.

Off-line folders and roaming folders are two major features of group policy. When a client logs off the network for the day, defined folders and files can be configured to move automatically to a local hard drive location. When the user comes back into work and logs onto the network once again, the off-line folders can be automatically synchronized and updated with the network copies. Folder policy can be deployed in one of two methods: basic, which redirects every user to the same network location; and advanced, where each group’s folders can be redirected to a specified network location. The environment variable %USERNAME% can also be used for more control.
Terminal Services

Terminal services will be the ultimate feature-set of Windows 2000. Most of us want email, word processing and, in general, very low processing power for the generic corporate application. The power applications such as games, CAD, or programming will still need their local processing but the rest of us don't. Terminal services execute on a remote server location and just the video, mouse, and keyboard changes are updated on your terminal server client. The TS client can be a standard PC; however, the emergence of instant-on net PCs, and ever-increasing bandwidth across corporate networks means that terminal services are a bonus for administration since the end-user can't change anything because all components are server based.

Active Directory Services Interface (ADSI)

ADSI is Microsoft's open offering to independent software vendors to write scripts and applications that will hook into the active directory service using this single set of interfaces. Any application that is ADSI compliant will be able to access Active Directory Service. The present directory-service structure is driven by a specific application programming interface (API) for communicating with AD clients and servers. ADSI objects are component object models (COMs)—objects that are either container or leaf objects as listed in Table 1.2.

<table>
<thead>
<tr>
<th>Container Objects</th>
<th>Leaf Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Namespaces</td>
<td>User</td>
</tr>
<tr>
<td>Country</td>
<td>Alias</td>
</tr>
<tr>
<td>Locality</td>
<td>Service</td>
</tr>
<tr>
<td>Organization</td>
<td>Print Queue</td>
</tr>
<tr>
<td>Organizational Unit</td>
<td>Print Device</td>
</tr>
<tr>
<td>Domain</td>
<td>File Share</td>
</tr>
<tr>
<td>Computer</td>
<td>Session</td>
</tr>
<tr>
<td>Group</td>
<td>Resource</td>
</tr>
</tbody>
</table>
Distributed File System

The distributed file system (DFS) used throughout Windows 2000 is hosted by Active Directory to track and automatically replicate files that can even be stored on different servers across the enterprise network, as shown in Figure 1.4. Service pack 1 allows this feature to work. The end result is that the reference point is not a drive letter but a folder.

The Metadirectory: The Present and the Future

At this point in time, directory services can be split into two basic levels, starting at the system-specific components and ending up at the complete global or enterprise level sometime in the future. The reality is that all countries will probably not be able fully to work together, but the design principles of X.500 that Windows 2000 are actually built upon are hoping for the best. The key term is the metadirectory. Right now we are approaching level one in terms of integration as shown in Figure 1.5.

- **Level 1**—Firewalls, email, routers, networks, and Web sites.
- **Level 2**—This will use a metadirectory service to store together all the separate objects found in Level 1 so they can all be centrally managed and modified worldwide. Cisco and Microsoft hope to be the glue for the network of the future.
**Level 3**—This is the metadirectory that would exist if all countries and people agreed to work together in the same manner. (Don’t hold your breath.)

In an effort to fill in the holes, independent hardware vendors Cisco, Lucent, 3Com, and many others are already manufacturing directory-enabled devices like hubs, switches, and servers that can communicate using the standard LDAP protocol employed by all directory services. Each directory-enabled device can take advantage of the ability to extend the directory database structure through the schema, adding its own object identifiers and attributes that then become part of the directory. These devices could then also be administered like any other object stored in AD.

### Under the Hood: Active Directory Structure

Active Directory is structured as a directory service used as unique identification of all users and resources found across an internetwork, linking them together. The Manhattan Yellow Pages, or the Who’s Who Internet Listing are two examples of a public directory services that can be used in
an alphabetical name format to search for and retrieve information. These existing Internet services use DNS (Domain Name System) as the locator mechanism. Each Windows 2000 server that implements the Active Directory service also offers shared network resources to any Windows 2000 Professional Client coincidentally using DNS as the locator service. Active Directory includes both the directory—the location(s) where the network resources are stored—and the network services. Network services are the driving force for Windows 2000, executing in the background to provide network support and access to the Active Directory resources. The Active Directory database is a central database that looks the same to the clients in Japan or in Toronto as shown in Figure 1.6.

All resources are stored in the Active Directory database as objects and then populated (or replicated) throughout the network to every domain controller running Windows 2000 Server and Active Directory in the tree or forest.

**NOTE**

*Exchange Server 5.5 was the initial model for the Active Directory database.*

**Active Directory Single Point of Access**

With a single point of access required for both user access and administration, an administrator with the appropriate permissions can log onto the network anywhere and administer all objects and network resources
Understanding Active Directory

as shown in Figure 1.7. The active directory database is also fully replicated; any changes made to the Active Directory database will automatically update all other replicas through multi-master replication.

Figure 1.7
Using ADUC, administration can be performed on all domains in the forest.

Scalability

The architecture of AD is modeled after Exchange Server 5.5’s directory structure and storage engine. Each AD structure can support database stores that can contain millions of objects per forest.

Integration

Windows 2000 Server(s) (Server, Advanced Server, and Data Center Server) and Windows 2000 Professional are fully integrated once Active Directory is installed and all servers are running in native mode. The important concept to remember is that emerging software applications will integrate with the system information stored in the directory. All upcoming Windows 2000 NOS features will require and embrace Active Directory in one way or another, taking advantage of the security and the central storage.

Open Standards

Microsoft has wisely decided to embrace existing standards instead of creating a completely new propriety model. Active Directory is joined at
the hip with DNS, using it for its name system for locating network services like the global catalog server and domain controllers. It can also exchange data with any application that supports the LDAP protocol. Other common name formats are supported as shown in Table 1.3.

<table>
<thead>
<tr>
<th>Name Format</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 822</td>
<td>Uses the format of an email address username @domain</td>
</tr>
<tr>
<td>HTTP URL</td>
<td>Formatted in <a href="http://domain/path">http://domain/path</a></td>
</tr>
<tr>
<td>UNC</td>
<td>Uses the format \gk.com\doc\newfile.doc</td>
</tr>
<tr>
<td>RFC 1799</td>
<td>Uses the format LDAP://server.gk.com/CN=FirstnameLastname, OU=system, OU=division, DC=development</td>
</tr>
</tbody>
</table>

Multi-Master Replication

Since we know that there are multiple copies of the Active Directory database, we should ask how they stay up to date with each other. The answer is, multi-master replication that monitors and replicates any changes made to a Domain Controller’s replica of the database, then propagates the changes to all other domain controllers. Replication provides a large measure of fault tolerance and redundancy, plus distributed administration, since most changes need not be made at a specific DC, they can be performed at any DC in the tree or forest and replication will then update the other replication partners (other DCs that are also hosting the domain).

The Parts of the Active Directory Database

There are three main categories or parts to the database/schema of Active Directory:

1. **Resources**—all devices that are physically attached to the network, which can be accessed by a user at their computer including:
   - File server hard drives
   - Software applications
Understanding Active Directory

- Folder(s) containing data files
- Modem(s)
- IP addresses of network devices
- CD-ROM towers

In order for a resource to be accessible by any user at any location across the network, each network resource must also be shared.

2. **Services**—a service is a system network application running and executing in the background of the server and the workstation. In the Control Panel, the Services icon lists the executing network services. Most, if not all, installed network services directly relate to the resource that they are hosting, as shown in Table 1.4.

<table>
<thead>
<tr>
<th>Network Resource</th>
<th>Associated Network Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Application</td>
<td>Terminal server</td>
</tr>
<tr>
<td>IP Address</td>
<td>DHCP</td>
</tr>
<tr>
<td>Printer</td>
<td>Printer service</td>
</tr>
<tr>
<td>Folder</td>
<td>File service</td>
</tr>
<tr>
<td>Hard Drive Partition</td>
<td>File service</td>
</tr>
</tbody>
</table>

3. **User Accounts**—A current user account is obviously necessary before any object can be accessed in the Directory. The user must supply a valid logon ID and password before accessing a resource in the directory.

**Objects and Attributes**

Active Directory references everything viewable in the Windows Explorer shell by the word *object* or *attribute*.

The directory is a collection of these objects; printers, files, folders, fax servers, file servers, databases, computer systems, and user accounts, as shown in Figure 1.8.

When a user wants to access an object stored in the directory, the access can be controlled by layers of security that define the level of access allowed at a specific point in the directory. Each object can be searched and located either by the name of the object, or by the attributes of the
object itself. Suppose that a user in the accounting department wanted to find the location of clipart for a certain type of presentation; he could request, “All business clipart.” If a color laser printer were needed fast because the known local printer was on the fritz, the user could request, “Find all color printers on the main floor.”

These two examples illustrate the use of attributes. An attribute describes the named object; the more attributes, the easier it is to search for a specific object, as detailed in Figure 1.9.

What’s X.500?

X.500 is an industry-wide standard for the design of directory services; it was first defined and published in 1988 by the International...
Understanding Active Directory

Telecommunications Union (ITU). Design modifications were made in
1993, and the latest updates in 1997. The X.500 standard defines the
model to be used, complete with defined object classes and protocols.
The same standard is also published and maintained by another stan-
dards body, the ISO, which you may have heard of with regard to the
OSI network layers.

The defined X.500 protocols are the delivery mechanism for delivering
the required data stored in the directory itself.

- Directory Access Protocol (DAP)
- Directory Service Protocol (DSP)
- Directory Information Shadowing Protocol (DISP)
- Directory Operational Binding Management Protocol (DOP)
- Lightweight Directory Access Protocol (LDAP)

The above protocols are defined by X.500; however, if they were all
implemented at the same time they would consume a ton of bandwidth.
The only one currently used by Active Directory is LDAP.

Lightweight Directory Access Protocol

Lightweight Directory Access Protocol (LDAP) is the protocol used to
add, modify, delete, query, and retrieve data stored in Active Directory.
LDAP provides the basic communication service of locating information
in the directory. It defines how a directory client can access a directory
server and access and share directory data. LDAP is used to store and
retrieve information from Active Directory that could be distributed
across multiple Windows 2000 servers containing multiple system serv-
ces and software applications.

It also puts in place a “platform-neutral” common structure for infor-
mation services required by today’s network operating systems and
database applications. It is used to store and retrieve information
for other Windows 2000 system services, such as DNS, DHCP, and
kerberos, which is stored in the directory.
LDAP Components

Client-to-Server

By holding common system information used by Windows 2000, LDAP simplifies the management of system information and also greatly reduces unnecessary duplication, as shown in Figure 1.10.

![Figure 1.10 LDAP and client-to-server communications.](image)

The client-to-server communication allows a user-installed application to make contact with a Windows 2000 server running Active Directory and to create, retrieve, modify, and delete data records.

Server-to-Server

Server-to-server communications define how servers share the contents of the directory tree and how they perform updates and replication amongst themselves as shown in Figure 1.11.

Most vendors, including Microsoft, deploy the client-to-server communications in a standard format. However, for Windows 2000 the server-to-server communications is a Microsoft mix of the proposed LDAP standard plus their own enhancements.

LDAP and Data Retrieval

LDAP is a protocol to retrieve requested data; it can be used to retrieve information from the directory in three distinct methods:

1. As a service protocol—Used by different applications or user requests to retrieve the desired information they require. For exam-
Understanding Active Directory

Figure 1.11
LDAP and server-to-server communications.


ple, a user creates a query, which is then sent to a search engine; the query is matched against an LDAP server, and this points to the place where the actual data are located.

2. As a application data exchange interface—Used by one software application to exchange data with another. For example, a Lotus Notes database can store records into an LDAP server so that an LDAP-supported client such as a Microsoft Outlook client can then retrieve it.

3. As a system service protocol—Used by the operating system to communicate information between its different components. For example, an LDAP server can contain the access rights of user accounts that are referenced by the login system and by the installed file system.
The LDAP Functional Model

The LDAP structure is based on six functional models, as shown in Figure 1.12.

The Naming Model
This model details how the tree of data is laid out so that you can build a tree-like representation of your entire organization. The tree is also called a DIT or a directory information tree; the Active Directory database file is called NTDS.DIT.

Each entry can correspond to any type of object—a file, a user account, a printer, a DNS resource record, a user's desktop preference, etc. Each node in the LDAP tree can be of any type and can have any number of child nodes of different varieties; there are no limitations. The naming model also defines how each entry can be accessed within the context of the entire directory information tree.

The Information Model
This model defines the attributes of each entry and the data types of these attributes. For example, for a user account there must be a definition of how the account name is represented; usually by a string of characters. The information model defines the supported data types (strings, binary data, Boolean values, integers, and floating point values).

The Functional Model
This model defines how data are accessed from the directory system and what LDAP commands can be used. The commands listed in Table 1.5
Understanding Active Directory

can be grouped into useful tasks such as: search and compare; add, delete, modify, and rename; and session control operations of bind, unbind, and abandon.

<table>
<thead>
<tr>
<th>LDAP Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search</td>
<td>Search the directory for matching directory objects</td>
</tr>
<tr>
<td>Compare</td>
<td>Compare one directory object to a set of data</td>
</tr>
<tr>
<td>Add</td>
<td>Add a new directory object</td>
</tr>
<tr>
<td>Modify</td>
<td>Modify a particular directory object</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete a particular directory object</td>
</tr>
<tr>
<td>Rename (Modify DN)</td>
<td>Rename or modify the distinguished name of a directory object</td>
</tr>
<tr>
<td>Bind</td>
<td>Start a session with an LDAP server</td>
</tr>
<tr>
<td>Unbind</td>
<td>End a session with an LDAP server</td>
</tr>
<tr>
<td>Abandon</td>
<td>Abandon an operation previously sent to an LDAP server</td>
</tr>
<tr>
<td>Extended</td>
<td>Extended operations command</td>
</tr>
</tbody>
</table>

The Security Model

This model defines how the directory is secured. Security can be established by sub-tree, by individual entry, or by an attribute within an entry. Under LDAP version 3, there is a common authentication framework known as the Simple Authentication and Security Layer (SASL); it works with Secure Sockets Layer (SSL), Transaction Layer Security, Secure MIME, Kerberos, and other security systems and protocols deployed in Windows 2000.

Active Directory, Netscape’s Directory Server, and NDS all use access control lists to provide security down to the directory objects and their attributes.

The Replication Model

This model is detailed for multi-server environments and defines how the directory information tree (DIT) is then replicated across servers.
The Management Model

This model defines how the tree is managed, either as a whole or in parts.

Both Active Directory and Novell’s NDS, allow you to assign specific jobs to administrators so that they can manage selected subdomains or organizational units (OU).

Active Directory Interoperability

Active Directory supports standards to provide a degree of interoperability with Microsoft products and third-party vendors.

Application Programming Interface

There are two APIs that can be used to access data stored in Active Directory: the Active Directory Service Interface (ADSI) and the LDAP C API.

ADSI

Active Directory Service Interface (ADSI) supports access to the objects stored in Active Directory. The new access method is also referred to as Component Object Model (COM) objects. COM began life as DDE and OLE for the object linking and embedding of data records across a network between clients and server locations.

The classic example is the manager who doesn’t know how to use computers that well, yet needs to know how the company is doing. A spreadsheet could be created for the manager so he or she can view it with summarized data pulled from different departments, and network locations throughout the company. Using OLE and DDE, a master spreadsheet could be linked with several other spreadsheets stored in different locations around the network. When the manager started up his spreadsheet, in the background his computer queried the other linked spreadsheets and provided an updated view of the company’s data.
Understanding Active Directory

OLE and DDE became the component object model (COM), allowing us to perform linking and embedding across the network and across the Internet, with security features added in.

COM then became DCOM, the Dynamic Component Object Model that allows for distributed applications in the active directory world of Windows 2000. With Active Directory, the choices of objects we can link with are much more than just spreadsheets or documents. Since everything in our 32-bit world is a defined object, we can link and embed with almost anything using COM as a communication tool. For COM access, “providers” are available for NetWare's NDS, Windows NT, Windows 95/98, LDAP, and the IIS metabase.

LDAP C API

The LDAP C API is a set of low-level C language APIs linked to the LDAP protocol. It is supported on all Windows platforms.

Synchronization

Directory synchronization services are provided with Windows 2000 for synchronizing Active Directory with the applications listed in Table 1.6.

<table>
<thead>
<tr>
<th>Application</th>
<th>Service Provided</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exchange 5.5</td>
<td>Active Directory Connector</td>
</tr>
<tr>
<td>Novel NDS</td>
<td>Services for NetWare 5.0</td>
</tr>
<tr>
<td>Lotus Notes</td>
<td>Supported when Exchange 2000 ships</td>
</tr>
<tr>
<td>GroupWise</td>
<td>Supported when Exchange 2000 ships</td>
</tr>
</tbody>
</table>

Backward Compatibility

Windows 2000 Servers are installed in a default mode called a mixed mode configuration that supports both Windows NT backup domain controllers and Windows 2000 domain controllers. Active Directory also
supports mixed mode for the purpose of both NT 4.0 servers and clients authenticating with the Windows NT LAN Manager authentication protocol. As a result, older Windows 95/98 and NT 4.0 clients can log on and access resources in a Windows 2000 Domain.

Microsoft Management Console (MMC)

Active Directory Tools are contained as a “snap-in” component. This new interface is used for all Windows 2000 tools and utilities and was first used in Internet Information Server 4.0 administration, as detailed in Figure 1.13.

Although the initial introduction to the MMC with the term “snap in” can be confusing, it’s actually a slick idea. Microsoft provides the empty shell of a utility called the “tool host” in official development lingo, and you fill in (snap in) each software component you want to use. The current version of MMC is 1.1 and can be activated by typing in MMC using the run option from the Start menu. After filling up your MMC console you can save your electronic toolbox as a console file with the .msc extension. This file can then be sent to other administrators at different locations for their use.
Administrative Solutions:
Creating Custom MMCs

Using the MMC is the only framework for using Windows 2000 utilities. There are several default MMC consoles included with Windows 2000 and others can be easily created from the UI. The extension used to save all MMC consoles is .MSC.

Creating an MMC for Local Administration

1. Click the Start button select Search and For Files or Folders.
   A. Search for files with the .MSC extension—you will find several consoles displayed.
   B. Double-click on the file devmgnt.msc—this will open the Device Manager.
   C. Try several others, for example compmgmt.msc and msinfo32.msc.
   D. Close the Search Results screen.

2. Click the Start button and select Run.
   A. Enter the text mmc and click OK. An empty console called Console1 will open.
   B. Maximize the Console1 and Console Root window.
   C. To check on the currently configured options click the Console button and select Options.
   The default operating mode when creating custom MMCs is Author mode. Take a few moments to select and read the description for the four console modes. You can choose the User mode that limits any changes from being made to MMCs that you create for other users or administrators.
   D. Make sure that the Author Mode is selected and then click OK.
   E. Click the Console button and select Save As....
   F. In the File Name: box type Disk Management and then click Save.
   G. From the Console menu select Exit.

3. Click Start, point to Programs | Administrative Tools and select Disk Management. Your MMC that was just created should load.
   A. From the top left corner, click the Console button and select Add/Remove Snap-in.... The Add/Remove Snap-in dialog box is displayed.
Creating Custom MMCs

Custom MMCs can be created for local users and then customized by modifying the available extensions of the snap-ins selected and by changing the mode of operation of the MMC.

1. Click the Start button and select Run.
   A. Enter the text mmc and click OK. An empty console called Console1 will open.
   B. Maximize the Console1 and Console Root window.
   C. Click the Console button and from the menu select Add/Remove Snap-in.
   D. From the Standalone tab click Add. The Add Standalone Snap-ins: appears.
   E. Click Add and select the snap-in Computer Management.
   F. From the Computer Management splash screen select Local computer and click Finish.
   G. Click Close and then OK.
   H. Expand Computer Management and note the preconfigured System Tools Device Manager and Local Users and Groups. By removing extensions, particular tree items can be removed, allowing you to create a custom console with limited functionality.

2. Click the Console button and from the menu, select Add/Remove Snap-in.
Understanding Active Directory

A. Click to highlight Computer Management (Local) and then select the **Extensions** tab.
B. Clear the Add all extensions check box, and from the Available extensions box clear the Device Manager and Local Users and Groups check box.
C. Click **OK**.
D. Expand Computer Management and note the removal of the System Tools Device Manager and Local Users and Groups.
E. Click the **Console** button and from the menu select **Options**.
F. From the Options splash screen change the Console mode to User mode–full access. Take a moment to review what this mode of operation removes.
G. Click the **Console** button and from the menu select **Save as**.
H. From the Save in dialog box select **Desktop**.
I. Save this custom console as Local Toolset.
J. Click the **Console** menu and select **Exit**.
K. Click the **Desktop** shortcut beside the Start button to display the Desktop.
L. Double-click the **Local Toolset** to launch the MMC.
M. Click the **Console** button and attempt to add a snap-in. Can you? You shouldn’t be able to.