Microsoft Windows Server 2008: An Overview

Solutions in this chapter:

- Server Manager
- Server Core
- Active Directory Certificate Services
- Active Directory Domain Services

☑ Summary
☑ Solutions Fast Track
☑ Frequently Asked Questions
Introduction

With the introduction of new revisions to Microsoft products—for example, Windows, Exchange, and Communications Server—we have seen a trend toward “roles” within each product, as opposed to the various products being an all-in-one type of solution (as with Exchange 2007), or being additional features that work as a snap-in, such as DNS in Windows 2003.

With earlier versions of Windows Server 2000 or 2003, an Active Directory server was just that—an Active Directory server. What we are trying to say here is that it was more-or-less an “all-or-nothing” deal when creating a domain controller in Windows 2003. Very little flexibility existed in the way a domain controller could be installed, with the exception of whether a domain controller would also be a global catalog server or flexible single master operation (FSMO) server.

The new roles in Windows Server 2008 provide a new way for you to determine how they are implemented, configured, and managed within an Active Directory domain or forest. The new roles (and the official Microsoft definitions) are as follows:

- **Read-only domain controller (RODC)** This new type of domain controller, as its name implies, hosts read-only partitions of the Active Directory database. An RODC makes it possible for organizations to easily deploy a domain controller in scenarios where physical security cannot be guaranteed, such as branch office locations, or in scenarios where local storage of all domain passwords is considered a primary threat, such as in an extranet or in an application-facing role.

- **Active Directory Lightweight Directory Service (ADLDS)** Formerly known as Windows Server 2003 Active Directory Application Mode (ADAM), ADLDS is a Lightweight Directory Access Protocol (LDAP) directory service that provides flexible support for directory-enabled applications, without the dependencies required for Active Directory Domain Services (ADDS). ADLDS provides much of the same functionality as ADDS, but does not require the deployment of domains or domain controllers.
Active Directory Rights Management Service (AD RMS)  
Active Directory Rights Management Services (AD RMS), a format and application-agnostic technology, provides services to enable the creation of information-protection solutions. AD RMS includes several new features that were available in Active Directory Rights Management Services (AD RMS). Essentially, AD RMS adds the ability to secure objects. For example, an e-mail can be restricted to read-only, meaning it cannot be printed, copied (using Ctrl + C, and so on), or forwarded.

Active Directory Federation Services (ADFS)  
You can use Active Directory Federation Services (ADFS) to create a highly extensible, Internet-scalable, and secure identity access solution that can operate across multiple platforms, including both Windows and non-Windows environments. Essentially, this allows cross-forest authentication to external resources—such as another company’s Active Directory. ADFS was originally introduced in Windows Server 2003 R.2, but lacked much of its now-available functionality.

These roles can be managed with Server Manager and Server Core. Discussing Server Core is going to take considerably longer, so let’s start with Server Manager.

Server Manager

Server Manager is likely to be a familiar tool to engineers who have worked with earlier versions of Windows. It is a single-screen solution that helps manage a Windows server, but is much more advanced than the previous version.

Using Server Manager to Implement Roles

Although we will be discussing Server Manager (Figure 1.1) as an Active Directory Management tool, it’s actually much more than just that.
In fact, Server Manager is a single solution (technically, a Microsoft Management Console [MMC]) snap-in that is used as a single source for managing system identity (as well as other key system information), identifying problems with servers, displaying server status, enabled roles and features, and general options such as server updates and feedback.

Table 1.1 outlines some of the additional roles and features Server Manager can be used to control:
Server Manager is enabled by default when a Windows 2008 server is installed (with the exception of Server Core). However, Server Manager can be shut off via the system Registry and can be re-opened at any time by selecting **Start | Administrative Tools | Server Manager**, or right-clicking **Computer** under the **Start** menu, and choosing **Manage** (Figure 1.2).

<table>
<thead>
<tr>
<th>Role/Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Directory Certificate Services</td>
<td>Management of Public Key Infrastructure (PKI)</td>
</tr>
<tr>
<td>Dynamic Host Configuration Server</td>
<td>Dynamic assignment of IP addresses to clients</td>
</tr>
<tr>
<td>Domain Name Service</td>
<td>Provides name/IP address resolution</td>
</tr>
<tr>
<td>File Services</td>
<td>Storage management, replication, searching</td>
</tr>
<tr>
<td>Print Services</td>
<td>Management of printers and print servers</td>
</tr>
<tr>
<td>Terminal Services</td>
<td>Remote access to a Windows desktop or application</td>
</tr>
<tr>
<td>Internet Information Server</td>
<td>Web server services</td>
</tr>
<tr>
<td>Hyper-V</td>
<td>Server virtualization</td>
</tr>
<tr>
<td>BitLocker Drive Encryption</td>
<td>Whole-disk encryption security feature</td>
</tr>
<tr>
<td>Group Policy Management</td>
<td>Management of Group Policy Objects</td>
</tr>
<tr>
<td>SMTP Server</td>
<td>E-mail services</td>
</tr>
<tr>
<td>Failover Clustering</td>
<td>Teaming multiple servers to provide high availability</td>
</tr>
<tr>
<td>WINS Server</td>
<td></td>
</tr>
<tr>
<td>Legacy NetBIOS name resolution</td>
<td></td>
</tr>
<tr>
<td>Wireless LAN Service</td>
<td>Enumerates and manages wireless connections</td>
</tr>
</tbody>
</table>
So, those are the basics of Server Manager. Now let’s take a look at how we use Server Manager to implement a role. Let’s take the IIS role and talk about using the Add Role Wizard to install Internet Information Services (IIS).
Tools & Traps...

Using the Add Role Wizard

Notice in Figure 1.1 that the Server Manager window is broken into three different sections:

- Provide Computer Information
- Update This Server
- Customize This Server

Under the Customize This Server section, click the Add Role icon. When the wizard opens, complete the following steps to install IIS onto the server.

1. Click the Add Roles icon.
2. At the Before You Begin window, read the information provided and then click Next.
3. From the list of server roles (Figure 1.3), click the check box next to Web Server (IIS) and then click Next.
4. If you are prompted to add additional required features, read and understand the features, and then click Add Required Features.
5. When you return to the Select Server Roles screen, click Next.
6. Read the information listed in the Introduction to Web Server (IIS) window and then click Next.
7. For purposes of this example, we will select all of the default Role Services and then click Next.
8. Review the Installation Summary Confirmation screen (Figure 1.4) and then click Install.
9. When installation is complete, click Close.
10. Notice that on the Server Manager screen, Web Server (IIS) is now listed as an installed role.
Figure 1.3 List of Server Roles

Figure 1.4 The Installation Summary Confirmation Screen
Configuring & Implementing...

Scripting vs. GUI
Sure, you can always use a wizard to implement a role, but you also have the option of using a script. Realistically speaking, it’s generally not the most efficient way to deploy a role for a single server, however. Unless you are going to copy and paste the script, the chance of error is high in typing out the commands required. For example, take the following IIS script syntax:

```
start /w pkmgr /iu:IIS-WebServerRole;IIS-WebServer;IIS-CommonHttpFeatures;
IIS-StaticContent;IIS-DefaultDocument;IIS-DirectoryBrowsing;IIS-HttpErrors;
IIS-HttpRedirect;IIS-ApplicationDevelopment;IIS-ASPNET;IIS-NetFxExtensibility;
IIS-ASP;IIS-CGI;IIS-ISAPIExtensions;IIS-ISAPIFilter;IIS-ServerSideIncludes;
IIS-HealthAndDiagnostics;IIS-HttpLogging;IIS-LoggingLibraries;IIS-Request
Monitor;IIS-HttpTracing;IIS-CustomLogging;IIS-ODBCLogging;IIS-Security;
IIS-BasicAuthentication;IIS-WindowsAuthentication;IIS-DigestAuthentication;
IIS-ClientCertificateMappingAuthentication;IIS-IISCertificateMappingAuthentication;
IIS-URLAuthorization;IIS-RequestFiltering;IIS-IPSecurity;IIS-
Performance;IIS-HttpCompressionStatic;IIS-HttpCompressionDynamic;IIS-WebServ
erManagementTools;IIS-ManagementConsole;IIS-ManagementScriptingTools;IIS-
ManagementService;IIS-IIS6ManagementCompatibility;IIS-Metabase;IIS-
WMICompatibility;IIS-LegacyScripts;IIS-LegacySnapIn;IIS-FTPPublishingService;
IIS-FTPServer;IIS-FTPManagement;WAS-WindowsActivationService;WAS-ProcessModel;
WAS-NetFxEnvironment;WAS-ConfigurationAPI
```

This script installs ALL of the IIS features, which may not be the preferred installation for your environment, and within the time it took to type it out, you may have already completed the GUI install!

Server Core
Server Core brings a new way not only to manage roles but also to deploy a Windows Server. With Server Core, we can say goodbye to unnecessary GUIs, applications, services, and many more commonly attacked features.
Using Server Core and Active Directory

For years, Microsoft engineers have been told that Windows would never stand up to Linux in terms of security simply because it was too darn “heavy” (too much) code, loaded too many modules (services, startup applications, and so on), and was generally too GUI heavy. With Windows Server 2008, Microsoft engineers can stand tall, thanks to the introduction of Server Core.

What Is Server Core?

What is Server Core, you ask? It’s the “just the facts, ma’am” version of Windows 2008. Microsoft defines Server Core as “a minimal server installation option for Windows Server 2008 that contains a subset of executable files, and five server roles.” Essentially, Server Core provides only the binaries needed to support the role and the base operating systems. By default, fewer processes are generally running.

Server Core is so drastically different from what we have come to know from Windows Server NT, Windows Server 2000, or even Windows Server 2003 over the past decade-plus, that it looks more like MS-DOS than anything else (Figure 1.5). With Server Core, you won’t find Windows Explorer, Internet Explorer, a Start menu, or even a clock! Becoming familiar with Server Core will take some time. In fact, most administrators will likely need a cheat sheet for a while. To help with it all, you can find some very useful tools on Microsoft TechNet at http://technet2.microsoft.com/windowsserver2008/en/library/e7e522ac-b32f-42e1-b914-53ccc78d18161033.mspx?mfr=true. This provides command and syntax lists that can be used with Server Core. The good news is, for those of you who want the security and features of Server Core with the ease-of-use of a GUI, you have the ability to manage a Server Core installation using remote administration tools.
Before going any further, we should discuss exactly what will run on a Server Core installation. Server Core is capable of running the following server roles:

- Active Directory Domain Services Role
- Active Directory Lightweight Directory Services Role
- Dynamic Host Configuration Protocol (DHCP)
- Domain Name System (DNS) Services Role
- File Services Role
- Hyper-V (Virtualization) Role
- Print Services Role
- Streaming Media Services Role
- Web Services (IIS) Role
Although these are the roles Server Core supports, it can also support additional features, such as:

- Backup
- BitLocker
- Failover Clustering
- Multipath I/O
- Network Time Protocol (NTP)
- Removable Storage Management
- Simple Network Management Protocol (SNMP)
- Subsystem for Unix-based applications
- Telnet Client
- Windows Internet Naming Service (WINS)

**Note**

Internet Information Server is Microsoft’s brand of Web server software, utilizing Hypertext Transfer Protocol to deliver World Wide Web documents. It incorporates various functions for security, allows for CGI programs, and also provides for Gopher and FTP servers.

**Note**

BitLocker Drive Encryption is an integral new security feature in Windows Server 2008 that protects servers at locations, such as branch offices, as well as mobile computers for all those roaming users out there. BitLocker provides offline data and operating system protection by ensuring that data stored on the computer is not revealed if the machine is tampered with when the installed operating system is offline.
The concept behind the design Server Core is to truly provide a minimal server installation. The belief is that rather than installing all the application, components, services, and features by default, it is up to the implementer to determine what will be turned on or off.

Installation of Windows 2008 Server Core is fairly simple. During the installation process, you have the option of performing a Standard Installation or a Server Core installation. Once you have selected the hard drive configuration, license key activation, and End User License Agreement (EULA), you simply let the automatic installation continue to take place. When installation is done and the system has rebooted, you will be prompted with the traditional Windows challenge/response screen, and the Server Core console will appear.

### Configuring & Implementing...

#### Configuring the Directory Services Role in Server Core

So let’s put Server Core into action and use it to install Active Directory Domain Services. To install the Active Directory Domain Services Role, perform the following steps:

1. The first thing we need to do is set the IP information for the server. To do this, we first need to identify the network adapter. In the console window, type `netsh interface ipv4 show interfaces` and record the number shown under the `Idx` column.

2. Set the IP address, Subnet Mask, and Default Gateway for the server. To do this, type `netsh interface ipv4 set address name="<ID>" source=static address=<StaticIP> mask=<SubnetMask> gateway=<DefaultGateway>`. ID represents the number from step 1, `<StaticIP>` represents the IP address we will assign, `<SubnetMask>` represents the subnet mask, and `<Default Gateway>` represents the IP address of the server’s default gateway. See Figure 1.6 for our sample configuration.

Continued
3. Assign the IP address of the DNS server. Since this will be an Active Directory Domain Controller, we will set the DNS settings to point to itself. From the console, type `netsh interface ipv4 add dnsserver name="<ID>" address=<DNSIP> index=1.`. ID represents the number from step 1, and <StaticIP> represents the IP address of the DNS server (in this case, the same IP address from step 2).

So, here is where things get a little tricky. When installing the Directory Services role in a full server installation, we would simply open up a Run window (or a command line) and type in `DCPromo`. Then, we would follow the prompts for configuration (domain name, file location, level of forest/domain security), and then restart the system. Installing the role in Server Core isn’t so simple, yet it’s not exactly rocket science. In order to make this installation happen, we are going to need to configure an unattended installation file. An unattended installation file (see Figure 1.7) is nothing more than a text file that answers the questions that would have been answered during the DCPromo installation. So, let’s assume you have created the unattended file and placed it on a floppy disk, CD, or other medium, and then inserted it into the Server Core server. Let’s go ahead and install Directory Services:

1. Sign in to the server.
2. In the console, change drives to the removable media. In our example, we will be using drive E:, our DVD drive.
3. Once you have changed drives, type `dcpromo answer:\answer.txt`. `Answer.txt` is the name of our unattended file (see Figure 1.7).
4. Follow the installation process as it configures directory services. Once the server has completed the installation process, it will reboot automatically.
5. When the server reboots, you will have a fully functional Active Directory implementation!
**Figure 1.6** Setting an IP Address in Server Core

```bash
C:\Users\administrator>netsh interface ipv4 show interfaces

<table>
<thead>
<tr>
<th>Idx</th>
<th>Met</th>
<th>MTU</th>
<th>State</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>10</td>
<td>1500</td>
<td>connected</td>
<td>Local Area Connection</td>
</tr>
<tr>
<td>1</td>
<td>50</td>
<td>4294967295</td>
<td>connected</td>
<td>Loopback Pseudo-Interface 1</td>
</tr>
</tbody>
</table>

C:\Users\administrator>netsh interface ipv4 set address name="2" source=static address=192.168.2.97 mask=255.255.255.0 gateway=192.168.1.1

C:\Users\administrator>netsh interface ipv4 show address

Configuration for interface "Local Area Connection"
- DHCP enabled: No
- IP Address: 192.168.2.97
- Subnet Prefix: 192.168.2.0/24 (mask 255.255.255.0)
- Default Gateway: 192.168.1.1
- Gateway Metric: 1
- InterfaceMetric: 10

Configuration for interface "Loopback Pseudo-Interface 1"
- DHCP enabled: No
- IP Address: 127.0.0.1
- Subnet Prefix: 127.0.0.0/8 (mask 255.0.0.0)
- InterfaceMetric: 50

C:\Users\administrator>
```

**Figure 1.7** Installing Directory Services in Server Core

```bash
[CCInstall]
AdministratorPassword = p@ssw0rd
CreateOnJoin = Create
DomainNetBiosName = uccentral
NewNameFromNtName = uccentral.adc
RebootOnSuccess = Yes
ReplicaOrNonDomain = Domain
SIdName = "Lab"
TreeOrChild = Tree
```
Uses for Server Core

A Windows Server 2008 Core Server Installation can be used for multiple purposes. One of the ways that Server Core can be used is to provide a minimal installation for DNS. You can manipulate, manage, and configure DNS servers through the various Windows Server 2008 DNS Graphical User Interfaces (GUIs)—DNS Manager and the Server Manager tool.

However, there are no GUIs provided with Windows Server 2008 Core Server. There are a number of advantages to running DNS within Server Core, including:

- **Smaller Footprint.** Reduces the amount of CPU, memory, and hard disk needed.
- **More Secure.** Fewer components and services running unnecessarily.
- **No GUI.** No GUI means that users cannot make modifications to the DNS databases (or any other system functions) using common/user–friendly tools.

If you are planning to run DNS within a Server Core install, there a number of steps you must perform prior to installation. The first step we must take is to set the IP information of the server. To configure the IP addressing information of the server follow these steps:

1. Identify the network adapter. In the console window, type `netsh interface ipv4 show interfaces` and record the number shown under the `Idx` column.
2. Set the IP address, Subnet Mask, and Default Gateway for the server. To do this, type `netsh interface ipv4 set address name="<ID>" source=static address=<StaticIP> mask=<SubnetMask> gateway=<DefaultGateway>`. ID represents the interface number from step 1, `<StaticIP>` represents the IP address we will assign, `<SubnetMask>` represents the subnet mask, and `<Default Gateway>` represents the IP address of the server’s default gateway. See Figure 1.8 for our sample configuration.
3. Assign the IP address of the DNS server. If this server were part of an Active Directory domain and replicating Active-Directory integrated zones (we will discuss those next), we would likely point this server to another AD-integrated DNS server. If it is not, we would point it to another external DNS server—commonly the Internet provider of your company. From the console, type `netsh interface ipv4 add dnsserver name="<ID>" address=<DNSIP> index=1`. ID represents the number from step 1, <StaticIP> represents the IP address of the DNS server.

   Once the IP address settings are completed—you can verify this by typing `ipconfig /all`—we can install the DNS role onto the Core Server installation.

4. To do this, from the command line type `start /w ocsetup DNS-Server-Core-Role`.

5. To verify that the DNS Server service is installed and started, type `NET START`. This will return a list of running services.
6. Next, we can use the `dnscmd` command line utility to manipulate the DNS settings. For example, you can type `dnscmd /enumzones` to list the zones hosted on this DNS server.

7. We can also change all the configuration options that we modified in the GUI section earlier by using the `dnscmd /config` option. For example, we can enable BIND secondaries by typing `dnscmd <servername> /config /bindsecondaries 1`. You can see the results in Figure 1.9.

**Figure 1.9 Using the `dnscmd` Utility**

![Using the `dnscmd` Utility](image)

There are many, many more things you can do with the `dnscmd` utility. For more information on the `dnscmd` syntax, visit [http://technet2.microsoft.com/WindowsServer/en/library/d652a163-279f-4047-b3e0-0c468a4d69f31033.mspx](http://technet2.microsoft.com/WindowsServer/en/library/d652a163-279f-4047-b3e0-0c468a4d69f31033.mspx).

**Active Directory Certificate Services**

In PKI, a digital certificate is a tool used for binding a public key with a particular owner. A great comparison is a driver’s license. Consider the information listed on a driver’s license:

- Name
- Address
- Date of birth
- Photograph
- Signature
- Social security number (or another unique number such as a state issued license number)
- Expiration date
- Signature/certification by an authority (typically from within the issuing state’s government body)

The information on a state license photo is significant because it provides crucial information about the owner of that particular item. The signature from the state official serves as a trusted authority for the state, certifying that the owner has been verified and is legitimate to be behind the wheel of a car. Anyone, like an officer, who wishes to verify a driver’s identity and right to commute from one place to another by way of automobile need only ask for and review the driver’s license. In some cases, the officer might even call or reference that license number just to ensure it is still valid and has not been revoked.

A digital certificate in PKI serves the same function as a driver’s license. Various systems and checkpoints may require verification of the owner’s identity and status and will reference the trusted third party for validation. It is the certificate that enables this quick hand-off of key information between the parties involved.

The information contained in the certificate is actually part or the X.509 certificate standard. X.509 is actually an evolution of the X.500 directory standard. Initially intended to provide a means of developing easy-to-use electronic directories of people that would be available to all Internet users, it became a directory and mail standard for a very commonly known mail application: Microsoft Exchange 5.5. The X.500 directory standard specifies a common root of a hierarchical tree although the “tree” is inverted: the root of the tree is depicted at the “top” level while the other branches—called “containers”—are below it. Several of these types of containers exist with a specific naming convention. In this naming convention, each portion of a name is specified by the abbreviation of the object type or a container it represents. For example, a \textit{CN=} before a username represents it is a “\textit{common name}”, a \textit{C=} precedes a “\textit{country}”, and an \textit{O=} precedes “\textit{organization}”. These elements are worth remembering as they will appear not only in discussions about X.500 and X.509, but they are ultimately the basis for the scheme of Microsoft’s premier directory service, Active Directory.
X.509 is the standard used to define what makes up a digital certificate. Within this standard, a description is given for a certificate as allowing an association between a user’s distinguished name (DN) and the user’s public key. The DN is specified by a naming authority (NA) and used as a unique name by the certificate authority (CA) who will create the certificate. A common X.509 certificate includes the following information (see Table 1.2 and Figures 1.10 and 1.11):

<table>
<thead>
<tr>
<th>Item</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial Number</td>
<td>A unique identifier</td>
</tr>
<tr>
<td>Subject</td>
<td>The name of the person or company that is being identified, sometimes listed as “Issued To”.</td>
</tr>
<tr>
<td>Signature Algorithm</td>
<td>The algorithm used to create the signature.</td>
</tr>
<tr>
<td>Issuer</td>
<td>The trusted authority that verified the information and generated the certificate, sometimes listed as “Issued By”.</td>
</tr>
<tr>
<td>Valid From</td>
<td>The date the certificate was activated.</td>
</tr>
<tr>
<td>Valid To</td>
<td>The last day the certificate can be used.</td>
</tr>
<tr>
<td>Public Key</td>
<td>The public key that corresponds to the private key.</td>
</tr>
<tr>
<td>Thumbprint Algorithm</td>
<td>The algorithm used to create the unique value of a certificate.</td>
</tr>
<tr>
<td>Thumbprint</td>
<td>The unique value of every certificate, which positively identifies the certificate. If there is ever a question about the authenticity of a certificate, check this value with the issuer.</td>
</tr>
</tbody>
</table>
**Figure 1.10** A Windows Server 2008 Certificate Field and Values
In Active Directory and Windows Server 2008, Certificate Services allow administrators to establish and manage the PKI environment. More generally, they allow for a trust model to be established within a given organization. The trust model is the framework that will hold all the pieces and components of the PKI in place. Typically, there are two options for a trust model within PKI: a single CA model and a hierarchical model. The certificate services within Windows Server 2008 provide the interfaces and underlying technology to set up and manage both of these types of deployments.
Configuring a Certificate Authority

By definition, a certificate authority is an entity (computer or system) that issues digital certificates of authenticity for use by other parties. With the ever increasing demand for effective and efficient methods to verify and secure communications, our technology market has seen the rise of many trusted third parties into the market. If you have been in the technology field for any length of time, you are likely familiar with many such vendors by name: VeriSign, Entrust, Thawte, GeoTrust, DigiCert, and GoDaddy are just a few.

While these companies provide an excellent and useful resource for both the IT administrator and the consumer, companies and organizations desired a way to establish their own certificate authorities. In a third-party, or external PKI, it is up to the third-party CA to positively verify the identity of anyone requesting a certificate from it. Beginning with Windows 2000, Microsoft has allowed the creation of a trusted internal CA—possibly eliminating the need for an external third party. With a Windows Server 2008 CA, the CA verifies the identity of the user requesting a certificate by checking that user’s authentication credentials (using Kerberos or NTLM). If the credentials of the requesting user check out, a certificate is issued to the user. When the user needs to transmit his or her public key to another user or application, the certificate is then used to prove to the receiver that the public key inside can be used safely.

Certificate Authorities

Certificates are a way to transfer keys securely across an insecure network. If any arbitrary user were allowed to issue certificates, it would be no different than that user simply signing the data. In order for a certificate to be of any use, it must be issued by a trusted entity—an entity that both the sender and receiver trust. Such a trusted entity is known as a Certification Authority (CA). Third-party CAs such as VeriSign or Entrust can be trusted because they are highly visible, and their public keys are well known to the IT community. When you are confident that you hold a true public key for a CA, and that public key properly decrypts a certificate, you are then certain that the certificate was digitally signed by the CA and no one else. Only then can you be positive that the public key contained inside the certificate is valid and safe.

In the analogy we used earlier, the state driver’s licensing agency is trusted because it is known that the agency requires proof of identity before issuing a driver’s license. In the same way, users can trust the certification authority because they know
it verifies the authentication credentials before issuing a certificate. Within an organization leveraging Windows Server 2008, several options exist for building this trust relationship. Each of these begins with the decisions made around selecting and implementing certificate authorities. With regard to the Microsoft implementation of PKI, there are at least four major roles or types of certificate authorities to be aware of:

- Enterprise CA
- Standard CA
- Root CA
- Subordinate CA

Believe it or not, beyond this list at least two variations exist: intermediate CAs and leaf CAs, each of which is a type of subordinate CA implementation.

**Standard vs. Enterprise**

An enterprise CA is tied into Active Directory and is required to use it. In fact, a copy of its own CA certificate is stored in Active Directory. Perhaps the biggest difference between an enterprise CA and a stand-alone CA is that enterprise CAs use Kerberos or NTLM authentication to validate users and computers before certificates are issued. This provides additional security to the PKI because the validation process relies on the strength of the Kerberos protocol, and not a human administrator. Enterprise CAs also use templates, which are described later in this chapter, and they can issue every type of certificate.

There are also several downsides to an enterprise CA. In comparison to a stand-alone CA, enterprise CAs are more difficult to maintain and require a much more in-depth knowledge about Active Directory and authentication. Also, because an enterprise CA requires Active Directory, it is nearly impossible to remove it from the network. If you were to do so, the Directory itself would quickly become outdated—making it difficult to resynchronize with the rest of the network when brought back online. Such a situation would force an enterprise CA to remain attached to the network, leaving it vulnerable to attackers.

**Root vs. Subordinate Certificate Authorities**

As discussed earlier, there are two ways to view PKI trust models: single CA and hierarchical. In a single CA model PKIs are very simplistic; only one CA is used within the infrastructure. Anyone who needs to trust parties vouched for by the CA is given
the public key for the CA. That single CA is responsible for the interactions that ensue when parties request and seek to verify the information for a given certificate.

In a hierarchical model, a root CA functions as a top-level authority over one or more levels of CAs beneath it. The CAs below the root CA are called subordinate CAs. Root CAs serve as a trust anchor to all the CA’s beneath it and to the users who trust the root CA. A trust anchor is an entity known to be trusted without requiring that it be trusted by going to another party, and therefore can be used as a base for trusting other parties. Since there is nothing above the root CA, no one can vouch for its identity; it must create a self-signed certificate to vouch for itself. With a self-signed certificate, both the certificate issuer and the certificate subject are exactly the same. Being the trust anchor, the root CA must make its own certificate available to all of the users (including subordinate CAs) that will ultimately be using that particular root CA.

Hierarchical models work well in larger hierarchical environments, such as large government organizations or corporate environments. Often, a large organization also deploys a Registration Authority (RA, covered later in this chapter), Directory Services and optionally Timestamping Services in an organization leveraging a hierarchical approach to PKI. In situations where different organization are trying to develop a hierarchical model together (such as post acquisition or merger companies or those that are partnered for collaboration), a hierarchical model can be very difficult to establish as both parties must ultimately agree upon a single trust anchor.

When you first set up an internal PKI, no CA exists. The first CA created is known as the root CA, and it can be used to issue certificates to users or to other CAs. As mentioned above, in a large organization there usually is a hierarchy where the root CA is not the only certification authority. In this case, the sole purpose of the root CA is to issue certificates to other CAs in order to establish their authority.

Any certification authority that is established after the root CA is a subordinate CA. Subordinate CAs gain their authority by requesting a certificate from either the root CA or a higher level subordinate CA. Once the subordinate CA receives the certificate, it can control CA policies and/or issue certificates itself, depending on your PKI structure and policies.

Sometimes, subordinate CAs also issue certificates to other CAs below them on the tree. These CAs are called intermediate CAs. Is most hierarchies, there is more than one intermediate CA. Subordinate CAs that issue certificates to end users, server, and other entities but do not issue certificates to other CAs are called leaf CAs.
Certificate Requests

In order to receive a certificate from a valid issuing CA, a client—computer or user—must request a certificate from a CA.

There are three ways that this request can be made:

- Autoenrollment
- Use of the Certificates snap-in
- Via a web browser

It is very likely that the most common method for requesting a certificate is autoenrollment, and we’ll discuss its deployment shortly. A client can also request a certificate by use of the Certificates snap-in. The snap-in, shown in Figure 1.12, can be launched by clicking Start | Run, and then typing in certmgr.msc and pressing Enter. Note that the Certificates snap-in does not appear in the Administrative Tools folder as the Certification Authority snap-in does after installing certificate services. Once you open the Certificate Snap-in, expand the Personal container, and then right-clicking the Certificates container beneath it. You can start the Certificate Request Wizard by choosing All Tasks | Request New Certificate..., as shown in Figure 1.12.

Figure 1.12 Certificates Snap-in
Next, you will receive the **Before You Begin** welcome screen, as shown in Figure 1.13. Click **Next**.

**Figure 1.13 Before You Begin**

Next to Welcome screen, the wizard prompts you to choose the certificate enrollment type. Figure 1.14 shows you the available options. You can choose only a type for which the receiving CA has a template. Once you choose an appropriate template, click **Enroll**.
Next to Certificate Enrollment screen, verify it reads, STATUS: Succeeded, as shown in Figure 1.15. Click **Finish** to complete the request.
The last method for requesting a certificate is to use a Web browser on the client machine. Note that if you use this option, IIS must be installed on the CA. In the next section, we show you the steps for requesting a certificate using a client machine in this manner.

**Tip**

The order of component installation can be important when dealing with CAs. If you install certificate services *before* you install IIS, a client will *not* be able to connect as in the exercise below until you run the following from the command line: `certutil -vroot`. This establishes the virtual root directories necessary for Web enrollment. Note also that you must have selected the Web enrollment support option during the certificate services installation procedure.
Request a Certificate from a Web Server

To request a certificate from a Web server, follow these steps:

1. On any computer for which you want to request a certificate, launch Internet Explorer (version 5.0 or later) by clicking **Start | Programs** or **All Programs | Internet Explorer**.

2. In the address bar, type `http://servername/certsrv`, where `servername` is the name of the issuing CA.

3. When the welcome screen appears, as shown in Figure 1.16, click **Request a Certificate**.

---

**Figure 1.16 Welcome Screen of the CA’s Web Site**
4. Click **User Certificate**, then **Submit** when the next screen appears.

5. When the **Certificate Issued** page appears, click **Install This Certificate**. Close the browser.

**Certificate Practice Statement**

As the use of X.509-based certificates continues to grow it becomes increasingly important that the management an organization of certificates be as diligent as possible. We know what a digital certificate is and what its critical components are, but a CA can issue a certificate for a number of different reasons. The certificate, then, must indicate exactly what the certificate will be used for. The set of rules that indicates exactly how a certificate may be used (what purpose it can e trusted for, or perhaps the community for which it can be trusted) is called a certificate policy. The X.509 standard defines certificate policies as “a named set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements.”

Different entities have different security requirements. For example, users want a digital certificate for securing e-mail (either encrypting the incoming messages signing outgoing mail), Syngress (as other Web vendors do) wants a digital certificate for their online store, etc. Every user will want to secure their information, and a certificate owner will use the policy information to determine if they want to accept a certificate.

It is important to have a policy in place to state what the appropriate protocol is for use of certificates—how they are requested, how and when they may be used, etc.—but it is equally as important to explain exactly how to implement those policies. This is where the Certificate Practice Statement (CPS) comes in. A CPS describes how the CA plans to manage the certificates it issues.

**Key Recovery**

Key recovery is compatible with the CryptoAPI architecture of Windows 2008, but it is not a necessary requirement. For key recovery, an entity’s private key must be stored permanently. The storage of private keys guarantees that critical information will always be accessible, even if the information should get corrupted or deleted. On the other hand, there is a security issue in the backup of the private keys. The archived private key should be used to impersonate the private key owner only if corruption occurs on your system.
Active Directory Domain Services

Active Directory Domain Services (AD DS) stores information about users, computers, and other devices on the network. AD DS is required to install directory-enabled applications. The following are improvements made in AD DS functionality:

- Auditing (log value changes that are made to AD DS objects and their attributes)
- Fine-grained password policies (functionality to assign a special password and account lockout policies for different sets of users)
- Read-only DCs (hosts a read-only partition of the AD DS database)
- Restartable AD DS (can be stopped so that updates can be applied to a DC)
- Database mounting tool (compare different backups, eliminating multiple restores)
- User interface improvements (updated AD DS Installation Wizard)

What Is New in the AD DS Installation?

AD DS has several new installation options in Windows Server 2008, including the following:

- RODC
- DNS
- Global Catalog (GC) servers

New OS installation options include Full Install and Core Server Install. The first thing you must do when adding a Windows Server 2008 DC to a Windows 2003 forest is to prepare the forest for the Windows 2008 server by extending the schema to accommodate the new server:

- To prepare the forest for Windows Server 2008 run the following command: `adprep /forestprep`.
- To prepare the domain for Windows Server 2008 run the following command: `adprep /domainprep`. 
It is recommended that you host the primary domain controller (PDC) emulator operations master role in the forest root domain on a DC that runs Windows Server 2008 and to make this server a GC server. The first Windows Server 2008 DC in the forest cannot be an RODC. Before installing the first RODC in the forest, run the following command: `adprep /rodcprep`.

Making sure the installation was successful, you can verify the AD DS installation by checking the following:

- Check the Directory Service log in Event Viewer for errors.
- Make sure the SYSVOL folder is accessible to clients.
- Verify DNS functionality.
- Verify replication.

To run `adprep /forestprep` you have to be a member of the Enterprise Admins and Schema Admins groups of Active Directory. You must run this command from the DC in the forest that has the Schema Master FSMO role. Only one Schema Master is needed per forest.

To run `adprep /domainprep` you have to be a member of the Domain Admins or Enterprise Admins group of Active Directory. You must run this command from each Infrastructure Master FSMO role in each domain after you have run `adprep /forestprep` in the forest. Only one Infrastructure Master is needed per domain.

To run `adprep /rodcprep` you have to be a member of the Enterprise Admins group of Active Directory. You can run this command on any DC in the forest. However, it is recommended that you run this command on the Schema Master.
Summary

The new features of Windows Server 2008 are very important because understanding where changes are implemented and understanding where features have been improved will help you understand why this technology acts the way it does. Knowing how to tell what hardware components are appropriate, and which operating systems are designed for which roles and functionalities, is critical when you are choosing a new server, or deciding whether an existing server is up to the new task.

Knowledge of key features such as Server Manager, Server Core, AD Certificate Services, and AD Domain Services can help improve the user experience, improve the system administrator experience, and improve organizational security.

Solutions Fast Track

Server Manager

☑ Server Manager is likely to be a familiar tool to engineers who have worked with earlier versions of Windows.

☑ Server Manager is a single solution that is used as a single source for managing identity and system information.

☑ Server Manager is enabled by default when a Windows 2008 server is installed.

Server Core

☑ Server Core brings a new way not only to manage roles but also to deploy a Windows Server.

☑ Server Core is a minimal server installation option for Windows Server 2008 that contains a subset of executable files, as well as five server roles.

☑ Microsoft defines Server Core as “a minimal server installation option for Windows Server 2008 that contains a subset of executable files, and five server roles.”
Active Directory Certificate Services

- In PKI, a digital certificate is a tool used for binding a public key with a particular owner. A great comparison is a driver’s license.
- X.509 is the standard used to define what makes up a digital certificate.
- The X.500 directory standard specifies a common root of a hierarchical tree although the “tree” is inverted: the root of the tree is depicted at the “top” level while the other branches—called “containers”—are below it.

Active Directory Domain Services

- With the release of Windows Server 2008, an Active Directory domain controller can be deployed in several new ways.
- Active Directory Domain Services (AD DS) stores information about users, computers, and other devices on the network.
- AD DS has several new installation options in Windows Server 2008, including RODC and DNS.
Frequently Asked Questions

Q: Can I add Windows Server 2008 to an existing Windows 2003 Active Directory environment?
A: Yes. Adding a Windows Server 2008 DC to a current Windows 2003 Active Directory domain will make no difference to the 2003 Active Directory domain. However, you must install a full installation. The first 2008 DC cannot be a 2008 RODC, as it will need a full installation of the 2008 DC from which to replicate data.

Q: I have closed the command prompt on the Server Core terminal, and now I only see a blue background and cannot get the command prompt window back up. How do I get the command prompt window back without restarting the server?
A: Press Ctrl + Alt + Delete on the keyboard, open the Task Manager, and from the File menu choose Run, then type cmd.exe and click OK. This will bring back the command prompt window.

Q: Is an upgrade from Windows 2000 Server to Windows Server 2008 supported?
A: No. Only an upgrade from Windows Server 2003 is possible.

Q: What is Network Access Protection?
A: Network Access Protection (NAP) deals with the problem of unhealthy computers accessing the organization’s network. NAP ensures that any computer that makes a connection to the network meets the requirements set out by the organization’s policies. This limits access to the network and provides remediation services.

Q: My evaluation copy of Windows Server 2008 is going to expire soon. Can I extend it?
A: You can extend the 30-day grace period up to three times, for a total of 120 days. Use the SLMgr.vbs script with the rearm parameter to reset the counter for another 30 days. You will need to perform this step every 30 days, up to the 120 days.
Q: I am trying to install a domain controller in a domain that is in a Windows 2003 functional level. Do I have to choose Windows 2008 functional level when I install Windows Server 2008?

A: No, the functional level can always be upgraded to Windows Server 2008 at a later date.

Q: I want to be able to assign different account lockout policies to different sets of objects within Active Directory. Is this possible?

A: Yes, AD DS has a new Fine-Grained Password Policy that can be applied.