Managing File and Folder Attributes

Windows 2000 files and folders have various properties, called *attributes*, some of which the administrator can use to provide a limited amount of data protection. Administrators or users assign many attributes to protect files and folders. Other file and folder attributes are automatically applied to system files during the installation of Windows 2000.

Windows 2000 File and Folder Attributes

There are seven Windows 2000 file and folder attributes. These file and folder attributes can be used on FAT, FAT32, and NTFS volumes, with the exception of the Compress, Encrypt, and Index attributes, which are available only on NTFS volumes.

Table 11-1 lists and describes the Windows 2000 file and folder attributes.

TABLE 11-1 Windows 2000 File and Folder Attributes

Attribute	Description
Archive	Indicates that the file or folder has been modified since the last backup. Is applied by the operating system when a file or folder is saved or created, and is commonly removed by backup programs after the file or folder has been backed up.
Compress	Indicates that Windows 2000 has compressed the file or folder. Is only available on NTFS volumes. Can be set by using Windows Explorer and by using the `compress` command-line utility. Can't be used in conjunction with the Encrypt attribute. In other words, a file can be encrypted or compressed, but not both. Is applied by administrators to control which files and folders will be compressed.
Encrypt	Indicates that Windows 2000 has encrypted the file or folder. Is only available on NTFS volumes. Can be set by using Windows Explorer and by using the `cipher` command-line utility. Can't be used in conjunction with the Compress attribute. Is applied by users and administrator to control which files and folders will be encrypted. Once a file or folder has been encrypted, only the user who encrypted the file or folder (or the Administrator) can open the file or folder and view its contents.

Continued

TABLE 11-1 *(continued)*

Attribute	Description
Hidden	Indicates that the file or folder can't be seen in a normal directory scan. Files or folders with this attribute can't be copied or deleted. Is automatically applied to various files and folders by Windows 2000 during installation. In addition, this attribute can be applied by administrators or users to hide and protect files and folders.
Index	Indicates that the file or folder is indexed by the Indexing Service. Is only available on NTFS volumes. Can be applied by administrators or users. Once this attribute has been applied to a file, users can use Windows Explorer to locate this file by searching for words or phrases contained in the file.
Read-only	Indicates that the file or folder can only be read—it can't be written to or deleted. Is often applied by administrators to prevent accidental deletion of application files.
System	Indicates that the file or folder is used by the operating system. Files or folders with this attribute can't be seen in a normal directory scan, and can't be copied or deleted. Can't be set by using Windows Explorer. You must use the `attrib` command-line utility to view or change this attribute. Is automatically applied to various files and folders by Windows 2000 during installation.

Using the Compress Attribute

The Compress attribute is typically used to conserve disk space. You should only use this attribute on files or folders that are infrequently accessed because accessing a compressed file or folder uses more processor time (on the server that contains the file) than accessing an uncompressed file or folder. If a large number of users access compressed files on a server, that server's performance may be degraded. You can only compress files and folders on NTFS volumes.

Using the Encrypt Attribute

The Windows 2000 feature that provides the capability of the Encrypt attribute is called the *Encrypting File System (EFS)*.You don't need to install EFS — it's installed by default and is transparent to users. When users assign

the Encrypt attribute, that's all there is to it. EFS does all the work. As stated previously, the Encrypt attribute is only available for files and folders on NTFS volumes.

The Encrypt attribute is normally applied by a user to protect sensitive data that should be accessed only by that user. It is typically applied at the folder level, because when applied to a folder, Windows 2000 encrypts all of the files in the folder. When applied to an individual file, this attribute must be reapplied each time the file is modified.

As stated previously, in a Windows 2000 domain environment, only the user who encrypted the file and the domain's Administrator account can open the file. On a local Windows 2000 computer that is not a member of a domain, only the local user who encrypted the file and the local Administrator account can open the file. The Administrator account, in both of these situations, is called the *recovery agent* because this account is assigned a special key that permits it to unencrypt (that is, recover) all encrypted files on the computer. If you want to designate additional recovery agents, you can use Group Policy to specify additional users (on the local computer, in an OU, or in an entire domain) who can open all encrypted files and folders.

The Encrypt and Compress attributes are mutually exclusive — you can use one or the other, but not both, on a file or folder.

Using the Read-only Attribute

The Read-only attribute is frequently used to prevent the accidental deletion of application files. When a user has the Write NTFS permission to a Read-only file or folder on an NTFS volume, the Read-only attribute takes precedence. The Read-only attribute must be removed before the file or folder can be modified or deleted. (I'll cover NTFS permissions a little later in this chapter.)

Assigning Attributes to Files or Folders

Any user who can access a file or folder on a FAT or FAT32 volume can modify that file or folder's attributes. Any user who has the Write NTFS permission (or any permission that includes the functionality of the Write permission) to a file or folder on an NTFS volume can modify that file or folder's attributes.

Most file and folder attributes can be changed or assigned by using Windows Explorer, as the following steps explain.

Connecting to Shared Folders

Users must connect to shared folders before they can access the resources they contain. In the next sections, I'll discuss how to connect to shared folders, including how to use common naming conventions, Windows Explorer, and the command line to connect to shared network resources.

Naming Conventions

A *naming convention* is an accepted method of identifying individual computers and their resources on the network.

The two common naming conventions used in Windows 2000 are the *universal naming convention (UNC)* and *fully qualified domain names (FQDNs)*.

A UNC name consists of a server name and a shared resource name in the following format:

```
\\Server_name\Share_name
```

In this format, *Server_name* represents the name of the server that the shared folder is located on, and *Share_name* represents the name of the shared folder.You can use a UNC name in this format to connect to a network share. For example, a shared folder named `Public` located on a server named SERVER1 would have the following UNC name:

```
\\SERVER1\Public
```

A UNC name can also specify the name of a subfolder within the share, the name of a file within the share, or the name of a file within a subfolder in the share using the following format:

```
\\Server_name\Share_name\Subfolder_name\File_name
```

You can use a UNC name in this format to access a specific folder or file, such as a data file on a remote server. For example, a data file named `Salaries.doc` in the `Payroll` folder located in a share named `HR` on a server named CORP would have the following UNC name:

```
\\CORP\HR\Payroll\Salaries.doc
```

An FQDN is a fancy term for the way computers are named and referenced on the Internet. FQDNs are often used on networks that use TCP/IP and DNS servers.The format of an FQDN is:

server_name.*domain_name*.*root_domain_name*

For example, the FQDN of a server named WOLF in a domain named `AlanCarter` in the `com` root domain would be: `wolf.alancarter.com`.

On Windows 2000 networks, you can replace the *Server_name* in a UNC with an FQDN. For example, to specify a share named `Books` on a server with an FQDN of `wolf.alancarter.com`, you could use: `\\wolf.alancarter.com\Books`. In addition, you can also replace the *Server_name* in a UNC with the IP address of the server.

Both UNC names and FQDNs can be used to connect to shared network resources in Windows Explorer and from the command line.

Shared folder permissions (commonly called *share permissions*) apply to the shared folder, its files, and subfolders (in other words, to the entire directory tree under the shared folder).

Share permissions are the only folder and file security available on a FAT or FAT32 volume (with the exception of file attributes), and only control over-the-network access to the share — local access is totally unrestricted on a FAT or FAT32 volume.

Table 11-2 lists and describes the Windows 2000 share permissions, from most restrictive to least restrictive.

TABLE 11-2 Windows 2000 Share Permissions

Permission	Description
Read	Permits a user to view a list of the share's contents (names of files and subfolders), to change the current folder to a subfolder of the share (sometimes called *traversing to subfolders*), to view data in files, and to run application files.
Change	Permits a user to perform all tasks included in the Read permission. In addition, permits a user to create files and subfolders within the share, to edit data files and save changes, and to delete files and subfolders within the share.
Full Control	Permits a user to perform all tasks included in the Change permission. In addition, permits a user to change NTFS permissions and to take ownership of files and folders (on shares located on NTFS volumes).

Share permissions are assigned by adding a user or group to the permissions list for the share. From an administrative standpoint, it's more efficient to add groups to the permissions list for a particular share than to add individual users. By default, the Everyone group is granted the Full Control permission to all newly created shared folders.

When assigning permissions to a share, you should consider assigning the most restrictive permission that still permits users to accomplish the tasks they need to perform. For example, on shares that contain applications, consider assigning the Read permission so that users can't accidentally delete application files.

You can use Windows Explorer or Computer Management to assign share permissions to shared folders on the local Windows 2000 computer. To assign share permissions to shared folders on remote computers, use Computer Management.

How User and Group Permissions Combine

It is not uncommon for a user to have permissions to a share and to be a member of multiple groups that have different permissions to that share. When this occurs, the user and group permissions are additive, and normally the *least restrictive* permission is the user's effective permission. For example, suppose a user is allowed the Read permission to a share, and a group that the user is a member of is allowed the Change permission to the share. The user's effective share permission is Change.

An exception to this rule occurs when a user is specifically *denied* a permission. Remember the Allow and Deny check boxes in the permissions list to the share? *A denied permission always overrides an allowed permission.* Whenever a user is specifically denied a permission, or is a member of a group that is specifically denied a permission, the user is denied that permission. If a user is allowed the Full Control permission, but is a member of a group that is denied the Full Control permission, the user is denied the Full Control permission to the share — in other words the user is denied all access to the share. For this reason, you should exercise care in denying a specific share permission to a user or group.

Here are two examples that illustrate how user and group share permissions combine.

Example 1

A user, RomanB, manages a shared folder named `SalesData` that contains Sales department data. RomanB is a member of three groups. Table 11-3 shows the `SalesData` share permissions assigned to RomanB and to the three groups of which he is a member.

TABLE 11-3 RomanB's Group Memberships and Share Permissions

User or Group	SalesData Share Permissions Assigned
RomanB	Allow—Full Control
Sales	Allow—Change
Everyone	Allow—Read
Domain Users	Allow—Read

Because share permissions are additive, RomanB's effective permission to the `SalesData` share is Full Control.

Example 2

Until recently, a user, PennyL, was a design analyst in the Marketing department. She has just been promoted to a management position in the Human Resources department. PennyL's network has a shared folder named `HRData` that contains Human Resources department data, including employee performance reviews. PennyL is a member of three groups. Table 11-4 shows the `HRData` share permissions assigned to the three groups of which PennyL is a member.

TABLE 11-4 PennyL's Group Memberships and Their HRData Share Permissions

Group	HRData Share Permissions Assigned
Managers	Allow—Read
HR	Allow—Change
Marketing	Deny—Full Control, Change, and Read

Because a denied permission always overrides an allowed permission, PennyL's effective permission to the `HRData` share is Deny – Full Control, Change, and Read. In effect, PennyL is specifically denied all access to the `HRData` share. The administrator should remove PennyL from the Marketing group so she can access the `HRData` share. Once PennyL is removed from the Marketing group, her effective permission to the `HRData` share will be Change.

Modifying a Share

After a share is created, you may want to modify its properties. You can assign multiple share names to a share, change the name of a share, or stop sharing a share.

Assigning Multiple Share Names to a Share

To assist different users in locating or recognizing a share, you can assign multiple names to the same share.

For example, a group of technical support engineers might routinely access a share called `CIM` (CompuServe Information Manager), and less technical personnel at a help desk might access this same share using the name `CompuServe`.

When you assign an additional name to a share, what you actually end up doing is creating a new share for the *same* network resource. When you create the new share *you must manually assign a new set of share permissions that apply only to new share*. The permissions from the original share are *not* automatically applied to the new share.

Changing a Share Name

Occasionally you may need to change a share name. Perhaps you want to assign a more intuitive share name for users, or you might need to comply with a newly established set of naming conventions. To change a share name, you must create a new share that uses the new name, configure permissions for the new share, and then remove the original share.

Administrative Shares

Every time you start Windows 2000 on a computer, Windows 2000 automatically creates several hidden shares that only members of the Administrators group (on the local computer) have permissions to access. These shared are referred to as *administrative shares* because they are used by Administrators to perform administrative tasks.

The Windows 2000 administrative shares are: `C$`, `D$`, `E$`, and so on (one share for the root of each hard disk volume on the computer); and a share named `Admin$`, which corresponds to the folder in which Windows 2000 is installed (*SystemRoot*). The `$` at the end of each administrative share causes the share to be hidden from users when they browse the network. If users are not specifically aware the share exists, they will not be able to connect to the hidden share. To connect to a hidden share, you have to type in the server name and share name in the Map Network Drive dialog box in Windows Explorer. You can't browse for hidden shares.

Administrative shares make it possible for an Administrator to connect to any hard disk on a computer and to access all of its files and folders, regardless of whether regular shares exist on that hard disk. In this way an Administrator can perform backup, restore, and other administrative functions on a Windows 2000 computer.

Any share can be configured as a hidden share by placing a `$` at the end of its share name. However, hiding a share by appending a `$` to the share name does *not* limit user access to the share. The hidden share retains it assigned share permissions. Only access to the hidden *administrative* shares is restricted, by default, to Administrators only.

If you don't want administrative shares available on a Windows 2000 computer, you can configure Windows 2000 to prevent the creation of administrative shares. To accomplish this, you can edit the registry. You can

edit the registry directly by using `Regedt32.exe`, or you can use the System Policy Editor to disable the creation of the hidden administrative shares. System Policy editor was covered in chapter 10.

Managing NTFS File and Folder Security

When files and folders are stored on an NTFS volume on a Windows 2000 computer, NTFS permissions can be assigned to provide a greater level of security than share permissions, because:

- NTFS permissions, unlike share permissions, can be assigned to individual files as well as folders. This gives an administrator a much finer level of control over shared files and folders than is possible by using only share permissions.
- NTFS permissions apply to local users as well as to users who connect to a shared folder over the network. This fills the large security loophole left when files and folders on FAT partitions are secured only by share permissions.

The following sections discuss NTFS permissions, including how they are assigned to files and folders, how NTFS permissions are applied, and how NTFS and share permissions interact.

NTFS Permissions

NTFS permissions, which can only be assigned to files and folders on NTFS volumes, protect data from unauthorized access when users connect to the share locally or over the network.

The standard Windows 2000 NTFS permissions that can be assigned to files and folders are listed and described in Table 11-5.

TABLE 11-5 Windows 2000 Standard NTFS Permissions

Permission	When Applied to a File, a User Is Able to . . .	When Applied to a Folder, a User Is Able To . . .
Read	View the file's contents, attributes, extended attributes, and permissions; and synchronize the file.	View a list of the folder's contents (names of files and subfolders), attributes, extended attributes, and permissions; and synchronize the folder.
Read & Execute	Perform all actions included in the Read permission. In addition, the user can execute the file (if it is an executable).	Perform all actions included in the Read permission. In addition, the user can change the current folder to a subfolder (traverse to subfolders).
List Folder Contents	This permission is not available on files.	Perform all actions included in the Read & Execute permission. This permission is not inheritable by files in a folder—it applies to the folder only.
Write	View the file's permissions and synchronize the file. In addition, the user can write data to the file, append data to the file, and change the file's attributes and extended attributes.	View the folder's permissions and synchronize the folder. In addition, the user can create files and subfolders in the folder, and can change the folder's attributes and extended attributes.
Modify	Perform all actions included in the Read & Execute and Write permissions. In addition, the user can delete the file.	Perform all actions included in the Read & Execute and Write permissions. In addition, the user can delete the folder.
Full Control	Perform all actions included in the Modify permission. In addition, the user can change the file's permissions and can take ownership of the file.	Perform all actions included in the Modify permission. In addition, the user can change the folder's permissions, take ownership of the folder, and delete files and subfolders within the folder.

Printing Terminology

Before you can fully understand printing with Windows 2000, you need to understand a couple of basic terms.

In the terminology associated with Windows 2000, the term *printer* does not represent a physical device that produces printed output. Rather, a printer is the software interface between the Windows 2000 operating system and the device that produces the printed output.

If you are used to working with a different operating system, such as NetWare or UNIX, you may be used to thinking of what Windows 2000 calls a printer as a combination of a print queue (or print spooler) plus a driver for the device that produces printed output.

In Windows 2000, the term *print device* (or *printing device*) refers to the physical device that produces printed output — what is more commonly referred to as a "printer."

Now that you have a grasp of basic Windows 2000 printing terminology, you're ready to move on to the nuts and bolts of printing in Windows 2000.

Windows 2000 Printing Overview

This section explains how Windows 2000 processes print jobs from the time the user selects Print in an application until the paper comes out of the print device. It also explains how enhanced metafiles (EMFs) are used in the network printing process.

The Print Process

Perhaps the easiest way to understand the Windows 2000 print process is to follow the steps that occur when a document is printed from an application in Windows 2000.

1. A user at a Windows 2000 computer starts the print process from an application, such as Word, usually by selecting Print from the File menu. This action creates the print job. (A *print job* is all of the data and commands needed to print a document.)
2. The application hands off the print job to the Graphics Device Interface (GDI).
3. The GDI initiates a request to the driver for the print device.
4. The driver for the print device converts the application's output (the print job) into either a Windows 2000 enhanced metafile (EMF) or into the RAW format. (The RAW format is ready to print, as is, and no further processing is required.) The driver then returns the converted print job to the GDI.
5. The GDI hands off the print job to the Windows 2000 spooler.
6. The Windows 2000 spooler determines whether the print device is managed by the computer that initiated the print job or by a network-connected computer.

   If the print device is managed by the local computer (the computer that initiated this print job), the spooler copies the print job to a temporary storage area on the local computer's hard disk.

   If the print device is managed by a network-connected computer, the spooler hands off the print job to the spooler on the network-connected computer. Then that spooler copies the print job to a temporary storage area on that computer's hard disk.
7. Once the spooler has copied the file to temporary storage, the print job is handed off to the local print provider on the computer that has the print job spooled to its hard disk.
8. The local print provider initiates a request to the print processor to perform any additional conversions needed on the file, such as converting from EMF to RAW. (Print jobs are always sent to the print device in the RAW format.) The print processor then returns the converted print job to the local print provider.
9. The local print provider adds a separator page to the print job (if it's configured to do so) and then hands off the print job to the print monitor.

10. The print monitor communicates directly with the print device and sends the ready-to-print print job to the print device.
11. The print device produces the printed document.

Figure 12-1 graphically illustrates the steps in the Windows 2000 print process. Notice that the spooler routes the print job to the local hard disk if the print device is managed by the local computer, and routes the print job to the spooler on the network-connected computer if the print device is managed by the network-connected computer.

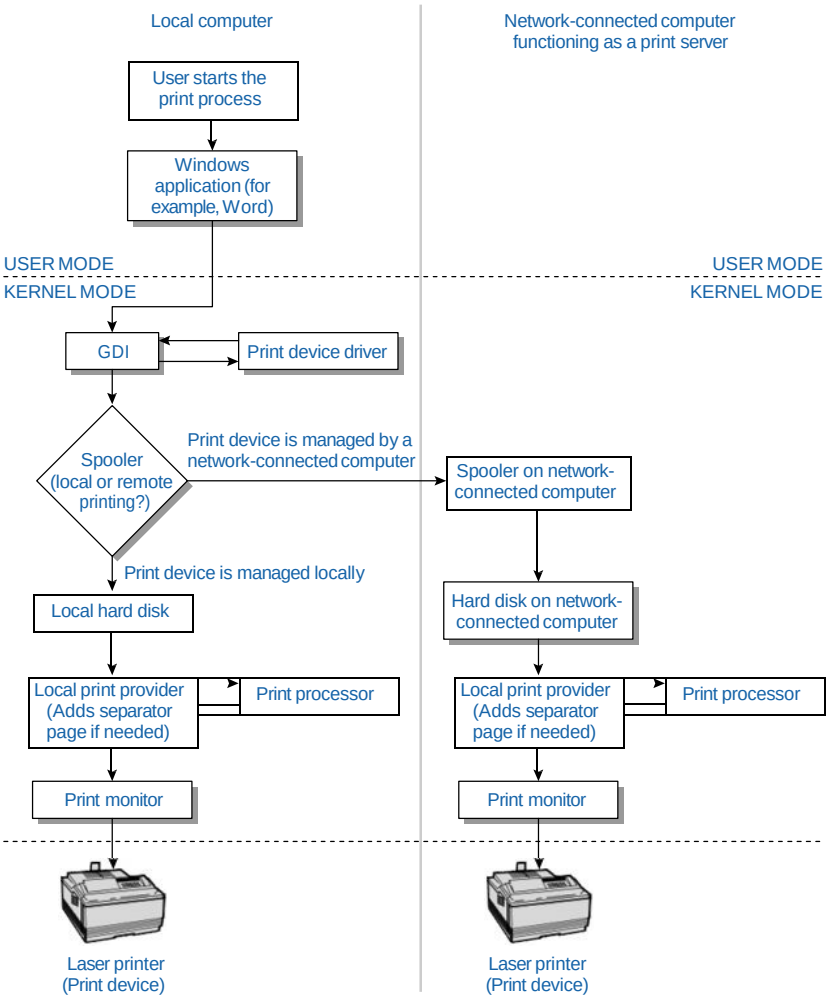

FIGURE 12-1 The Windows 2000 print process

Managing and Optimizing the Availability of User Data and System State Data

As a network administrator, it's your job to manage and optimize the availability of data on your network. In a nutshell, this means you have to secure the data on your company's network, protect it from loss, and ensure that it's always available when users need it. That's a tall order.

The data on your Windows 2000 network can be divided into two primary types: user data and System State data.

User data is a broad category that includes application files and folders, operating system files and folders, and user-created files and folders. In short, user data includes all files and folders on the Windows 2000 computer that aren't held open at all times by Windows 2000.

System State data includes various critical operating system files, folders, and databases. The actual components of System State data vary depending on the Windows 2000 operating system you're using and the services installed on that operating system. For all Windows 2000 computers, System State data includes the operating system boot files, the registry, and the COM+ Class Registration database. On a Windows 2000 Server computer that has Certificate Services installed, System State data also includes the Certificate Services database. Finally, on a Windows 2000 Server that is a domain controller, System State data also includes the Active Directory data store and the contents of the `SYSVOL` folder.

In this book, I've already discussed several ways you can manage and optimize the availability of your network's data, including using NTFS and permissions to restrict access to files and folders and using mirrored volumes and RAID-5 volumes to provide fault tolerance. Another important part of your overall fault tolerance plan is performing regular backups of data.

A tape backup is not a replacement for other fault tolerance methods, such as mirrored volumes and RAID-5 volumes. Tape backup is an additional safety precaution to use when other fault tolerance methods fail. I don't recommend that you rely solely on mirrored volumes, RAID-5

volumes, or tape backup. A comprehensive fault tolerance policy typically should include two or more of these strategies.

Backing Up User Data and System State Data

As I mentioned in the previous section, backing up data is an important part of your network fault tolerance plan. Planning and adhering to a regular backup schedule can make recovering from a corrupt file or a failed hard disk a straightforward, if somewhat painful, task. Failing to make regular backups of your system's critical data can be harmful (or even fatal) to your business, to your employment status, or both.

Always remember that a tape backup is your last line of defense against data loss. If the data on the tape is too old to be of value, or if it is corrupt, or if the tape has been damaged due to fire or other causes, then you have nothing. And having nothing is very hard to explain to upper management.

I can't stress enough the importance of carefully performing regular tape backups, and periodically testing the validity of those backups. Once you've experienced a partial or total disk failure, you'll never regret the time it takes you to perform backups again.

In the following sections I'll discuss what to back up, backup types, backup strategies, and how to use Backup to perform various tasks.

What to Back Up

Before you can create a backup strategy, you need to determine which data on your network will be backed up. I recommend that all network data be backed up regularly. This includes both user data and System State data.

In general, operating systems, applications, and System State data need to be backed up less frequently than user-created data files. You may find it sufficient to back up these types of data once a week, once a month,

even less often. An exception to this general rule is System State data on domain controllers. System State data on Windows 2000 domain controllers should be backed up fairly frequently because it contains the Active Directory data store.

Depending on the importance of your data, user-created data files can be backed up once a week, once a day, once an hour, or at any frequency that meets your organization's needs. When determining which files to back up and how often, ask yourself how much data you can really afford to lose. For example, if you decide to back up only once a week, can you afford to lose six days of sales information and other employee-created data?

Backup Types

Before I talk about the specific backup types, a short discussion on the archive attribute, and how the operating system and backup programs use this attribute, is in order.

The archive attribute is a marker that the operating system automatically assigns to all files and folders when they are first installed or created. Depending on the backup type, backup programs remove the archive attribute from a file or folder to indicate that the file or folder has been backed up. If a file or folder is modified after it is backed up, the operating system reassigns the archive attribute to it.

There are five standard types of backups you can perform:

- **Normal:** A normal backup backs up all selected files and folders. It removes the archive attribute from the backed up files and folders. A normal backup is a full, complete backup — it is the backbone of your backup plan or strategy.
- **Copy:** A copy backup backs up all selected files and folders. It does not remove or otherwise affect the archive attribute. The copy backup can be performed without disrupting the normal backup schedule, because it does not affect the archive attribute. You could use a copy backup to create an extra backup to store off-site.
- **Incremental:** An incremental backup backs up all selected files and folders that have changed since the last normal or incremental backup. An incremental backup removes the archive attribute from the backed up files and folders. An incremental backup is not cumulative — it contains only the changes made since the last normal or incremental backup. If a normal backup is performed

on Sunday, and incremental backups are performed Monday through Friday, Monday's incremental backup will contain all changes made to data on Monday, Tuesday's incremental backup will contain all changes made to data only on Tuesday, Wednesday's incremental backup will contain all changes made to data only on Wednesday, and so on. Because less data is backed up, an incremental backup takes less time to perform than a normal backup, and also takes less time to perform than a differential backup.

- **Differential:** A differential backup backs up all selected files and folders that have changed since the last normal backup. A differential backup does not remove the archive attribute from any files and folders. A differential backup is a cumulative backup since the last normal backup. Because the differential backup does not remove the archive attribute, if a normal backup is performed on Sunday, and differential backups are performed Monday through Friday, Monday's differential backup will contain all changes made to data on Monday; Tuesday's differential backup will contain all changes made to data on Monday and Tuesday; Wednesday's differential backup will contain all changes made to data on Monday, Tuesday, and Wednesday, and so on. A differential backup is often used in between normal backups, because it takes less time to perform a differential backup than a normal backup.
- **Daily:** A daily backup backs up all selected files and folders that have changed during the day the backup is made. It does not

Companies often use a combination of the standard backup types in their backup strategy.

Backup Strategies

There are a number of acceptable backup strategies, and three fairly common ones:

- **Perform a normal backup every day.** This is the most time-consuming of the three common strategies in terms of the time required to perform backups. However, should a restore be necessary, only the last normal backup is required, and restore time is greatly less than either of the other two strategies.

- **Perform a weekly normal backup and daily differential backups.** As the week progresses, the time required to perform the differential backups increases. However, should a restore be necessary, only two backup sets will be needed — the most recent normal backup, and the most recent differential backup. (This is because the most recent differential backup contains all files and folders that have changed since the last normal backup.) The restore can be accomplished relatively quickly.
- **Perform a weekly normal backup and daily incremental backups.** Incremental backups tend to take about the same amount of time each day, and are considered the fastest backup method. However, should a restore be necessary, multiple backup sets will be required — the most recent normal backup, and every incremental backup since the most recent normal backup. (This is because the incremental backups each contain different data and are not cumulative.) The restore will typically take more time than if a differential backup had been used.

When planning your backup strategy, the big trade-off you need to consider is time — the time it takes to perform backups versus the time it takes to restore data.

Security Considerations

When planning your company's backup strategy, there are a few security considerations to take into account:

- If the data is of a sensitive nature, consider physically securing the tape drive and the backup tapes in a locked room. While your server may require a password and permissions to access confidential data, when a backup tape is taken and restored on another server, your server's security measures are defeated.
- Consider rotating backup tapes to an off-site location. This can prevent or minimize data loss due to a single catastrophic event, such as a theft, fire, flood, or earthquake. Consider using a third-party company that will store your data tapes in a secure, climate-controlled environment.
- If you store backup tapes in a fireproof safe, remember that fireproof doesn't necessarily mean that heat or smoke can't destroy the data on magnetic tapes. Make sure the safe is capable of protecting magnetic media as well as papers and other important items.