Chapter 4

Files and File Management

In this chapter, you'll learn about:

- **File naming conventions**
- **Exchanging files across platforms**
- **File backups and file backup strategies**
- **Version control**
- **Basic asset management**

This chapter discusses several important but often-ignored topics of game graphics development: how to name, exchange, back up, maintain, and manage the files you create. Without understanding these issues and why they're necessary, your projects will suffer. However, if you take the time to appreciate them and apply what you learn here into your daily routine, your projects will go smoother and faster, and the work that you produce will be of higher quality. Sound good? Great, then read on.

File Naming Conventions

Once you determine the best graphic file format to store your game graphics in, you should start thinking about how to actually name the files you create. If you don't do this, you'll soon find yourself with lots of files on your hard drive and absolutely no idea of what they are or what they contain. Establishing a good file naming convention early on in your projects is a good practice and will definitely save you time and frustration later on.

However, before you start naming all of your files a certain way, you should make sure that you fully understand the various limitations imposed by your particular operating system as well as any operating system with which you're likely to exchange files. Unfortunately, DOS, Windows 95/98/NT/2000, Linux, and the Macintosh all have somewhat different rules for how they go about naming files. You need to know what these differences are if you plan on working with and exchanging files between these platforms.

DOS and Windows 3.1 File Naming Rules

- Both DOS and Windows 3.1 use what is known as an "8.3" file naming scheme. This includes eight alphanumeric characters, a dot, and then three more characters to hold the file's extension. This gives you a grand total of eleven characters to use for filenames or folders, which in the grand scheme of things isn't a whole lot.
- DOS and Windows 3.1 filenames aren't case sensitive; this means that filenames can be either upper- or lowercase.
- Filenames and directories can't include blank spaces or characters such as / \ : * ? < > . or |. All other characters available on your keyboard are legal for filenames, however.
- Neither DOS nor Windows 3.1 require you to include a file extension. However, it's a generally a good idea to do so since many applications on these systems use the file extension in order to identify what a file is.

Windows 95, 98, NT 4.0, and 2000 File Naming Rules

- Windows 95, 98, NT 4.0, and 2000 are all much more flexible than DOS or Windows 3.1 and can support filenames of up to 256 characters.
- Like their older relatives, you can't use characters such as / \ : * ? < > | or . in your filenames or folder names.
- Unlike DOS or earlier versions of Windows, spaces are okay to use.
- File extensions are also optional on these systems, but again, it's a good idea to include them because many Windows applications use this information to determine what different files are.
- Like DOS and Windows 3.1, the newer versions of Windows aren't case sensitive. For example, a filename like `arcade game graphics.bmp` is a perfectly legal filename under Windows 95, 98, NT 4.0, and 2000.

NOTE: Neither DOS nor Windows 3.1 like the longer filenames used by the newer Microsoft operating systems. They will read files created by these systems with no problem, but their filenames will be truncated after the seventh character just before the file extension. For example, the Windows 95 filename `spaceship1999.bmp` will appear as `spacesh~1.bmp` under DOS. Long filenames will display normally when using the DOS console on Windows 95, 98, NT 4.0, or 2000, however.

Macintosh File Naming Rules

- All Macintosh computers allow filenames and folder names to be up to 31 characters long.
- You can also use any character or combination of characters except for the colon (:).
- As with DOS and Windows, character case doesn't matter.
- Macintosh files don't require you to use file extensions. Unlike DOS or Windows, Macintosh applications don't rely on a file's extension to tell what it is. Rather, they use what is known as a *creator code* to identify different types of files and associate them with the applications they should work with. Every application on the Macintosh assigns files their own unique creator code. This allows you to do things such as launch the application that generated the file by simply double-clicking on its filename.

NOTE: When exchanging files and PC formatted disks with Macintosh computers running Mac OS 8.1 or lower, please be aware that Windows 95, 98, NT 4.0, and 2000 long filenames won't be preserved. These versions of the Mac OS will automatically truncate the filenames to adhere to the old, 8.3

naming standards. Mac OS 8.5 and above don't have this problem, however, and these systems will, in fact, preserve Windows long filenames.

Linux File Naming Rules

- Filenames and directories under Linux can be between 1 and 256 characters long.
- The following characters are legal for Linux filenames: a..z, A..Z, 0..9, _. The following characters have special meaning to Linux and really shouldn't be used in filenames or directories: < > ' " * { } [ ] ( ) ^ ! \ | & $? ~.
- Spaces aren't illegal in Linux filenames and directories, but they're not recommended. Use underscore (_) characters instead of spaces whenever possible.
- Do not start any file or directory name with a period (.), as this tells Linux to make it hidden from the system.
- Unlike the other platforms mentioned here, filenames are case sensitive under Linux. Therefore, Linux sees the files `stars.gif` and `Stars.gif` as two distinct files.

General File Naming Guidelines

The next step is to define the basic rules that identify the different files you create. Given that most of the graphics you design will have a specific function within a game, I suggest naming your files according to these basic guidelines:

- **Name your files logically**—This is a no-brainer. Always give your files logical, intelligent names in order to avoid any unnecessary confusion. For example, call the file that contains the spaceship sprites for level 5 of your game `spaceships.bmp` and not `ships.bmp`.

NOTE: This might cause compatibility problems with DOS systems so skip this step on this platform as necessary.

- **Consult with others**—Always check with your fellow developers and programmers when developing a file naming convention before starting a project. This will save you all time and grief later on should anyone involved have specific preferences, expectations, or needs regarding how graphic files should be named.
- **Be descriptive**—Try to be as descriptive as possible in naming your files but don't go overboard. For example, name the file that contains the alien spaceships for level 1 of your game `alien_spaceships_01.bmp`. Keep the filename restrictions of each platform in mind when naming your files. If you

intend to use multiple files of related images, you should number them sequentially, i.e., 00-99. When you run out of digits, use the letters AA through ZZ. For example, AA, AB, etc.

- **Use file extensions**—In addition to helping DOS, Windows, and Linux systems identify which program a particular file belongs to, file extensions also allow you to perform many powerful file manipulations on these systems. For example, you can decide to delete files that have a .PCX extension by typing del *.PCX or quickly copy all files to another drive that have a .BMP extension by typing copy C:\DATA *.BMP D:\BACK.
- **Adhere to the 8.3 naming scheme**—If cross-platform compatibility with DOS and Windows 3.1 systems is a concern, remember to restrict your filenames and directory names to the 8.3 naming rules described earlier in this section. Doing this will keep your filenames consistent even when used on different platforms.
- **Don't use spaces**—I suggest using underscore characters (_) in place of spaces even though the Macintosh and the current versions of Windows allow them in filenames. Doing this will give you more flexibility when performing file searches and wildcard file manipulations. It also allows you to maintain consistent filename compatibility with DOS and other operating systems including Linux.
- **Be consistent**—Once you establish a file naming scheme, stick to it. If you don't you're only asking for trouble as you'll soon experience problems finding, tracking, and maintaining your files.
- **Provide documentation**—You should always document your file naming scheme and distribute it to everyone who works with your files. The whole point of establishing your own file naming convention is wasted if you don't take the time to write out and define what your naming system actually means! Doing so will also avoid confusion regarding a given file's contents, especially when working with other people as part of a development team.

A Sample File Naming Scheme

This section provides an example of how you might opt to name your files in your own game projects.

The sample file naming scheme uses this format:

type-description-size.*extension*

Type:

This part of the filename indicates the type, or class, of file to which the image belongs. Please refer to Table 4-1 for a complete description of the possible image type prefixes.

TABLE 4-1: Possible Image Type Prefixes

Type Prefix	*Comments*
bkg	A background. This prefix is reserved for any image that will have one or more images displayed over it.
blk	An image block. This prefix is reserved for various image segments used for items such as status indicators and game text.
img	A generic image block. Typically, this prefix is used for title screens and other static image screens.
mnu	A menu element. Images that use this prefix can include everything from pre-made menu screens to individual button elements.
spr	A sprite image. This prefix should be used for any image that is to be animated within a game.

Description:

This is the description of the file in question.

As indicated in the earlier file naming guidelines, you should give your files logical and descriptive names. In general, try to limit your descriptions to less than 32 characters. Any more and the names become unwieldy to view and manage. Also, separate multiple word descriptions with underscore (_) characters and not spaces, such as `wilddog_running_left`.

NOTE: Limit your file descriptions accordingly in order to maintain compatibility with the Macintosh and other systems that place limits on the length of filenames.

Size:

This part of the filename contains the size of the file in pixel format such as 64x64 or 320x240.

This part of the filename is important since not all images will contain the same dimensions and there are many instances where the programmer needs exact image dimensions.

Extension:

This is the normal extension for a graphic file. Make sure you always include it so that the programmer (and other applications) will be able to instantly recognize its type.

TABLE 4-2: Example Asset Filenames

File	*Example Filename*
Game title screen	img-red_game_title-640x480.bmp
Game sprite	spr-green_dragon_moving_right-128x128.bmp
Game menu	mnu-play_button-120x40.gif
Game status indicator	blk-game_score_counter-160x20.gif

Managing and Organizing Directories

Directories (or folders, depending on your platform and nomenclature preference) allow you to store and organize groups of files much like a filing cabinet does. As such, directories can come in handy for managing the files you create in your game graphics projects.

NOTE: Directories are essentially files that contain other files. Therefore, all of the rules that apply to files will apply to directories as well.

To be effective and efficient, it's best to follow these simple rules:

- **Start fresh**—Always create a new directory for any project that is likely to produce more than one file. This will help you quickly organize and find files associated with a project. For example, you would create a directory called `JUMPER` to store the files for the new arcade game called "Jumper" on which you're going to start work.
- **Branch out**—Directories allow you to organize your files in a hierarchical, tree-like fashion. You can create special directories within directories called *subdirectories*. Take advantage of them in order to help categorize and better organize your files. For example, you might create a subdirectory called `BACKGRDS` to hold your background images and a subdirectory called `SPRITES` to hold your sprite screen.
- **Name smart**—Name your directories the same way you would name a file. That is, you should take your platform and its limitations into consideration (DOS, Windows, or the Macintosh), and use logical names, as shown in the previous example. Whenever possible, use descriptive directory names! This will help you find the files you want much faster.

Figure 4-1 shows an example of what your directories might look like in an actual project.

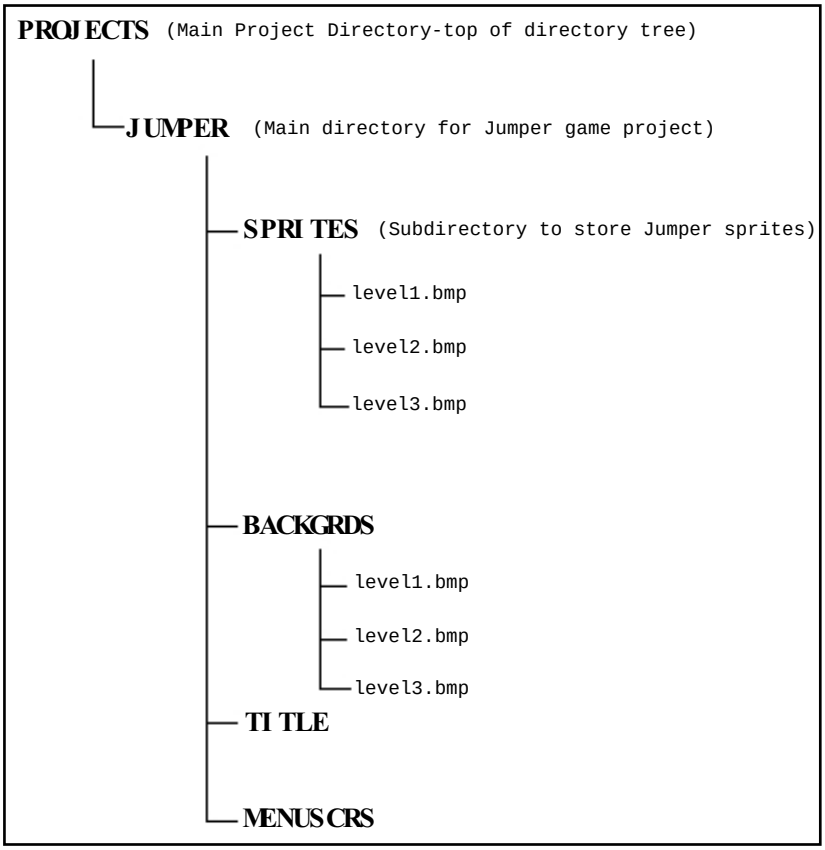

FIGURE 4-1: Example Game Project Directory

Exchanging Files Across Platforms

In a world where upwards of 90% of computers are running some version of DOS or Windows, exchanging files between different platforms is a lot easier than it used to be. However, it's still not a given, particularly with regard to non-PC systems such as the Macintosh or Linux. This being said, there are two basic levels of cross-platform compatibility that you should be worried about. These are:

- Disk compatibility
- File compatibility

Disk Compatibility

Mechanically speaking, there is really no different between how disks, be it a floppy, Zip, or CD-ROM, work across different platforms. Regardless of architecture, most computers manufactured since the late 1980s pretty much all use the same components, technology, and hardware. However, there's a huge difference between platforms in how information is recorded and stored. These differences become obvious whenever you try to exchange disks between different systems using their native formats. Without compatibility standards, such operations are usually doomed to failure.

Fortunately, there's one way to ensure almost universal disk compatibility: standardization! By this I mean that you should standardize on a single disk format for all of your disks. For obvious reasons, this standard disk format should be the DOS (FAT) format.

Macintosh Disk Compatibility with Different Platforms

- Format all floppies using the "DOS 1.4 MB" option from the Finder's Erase Disk command. This creates a 100% compatible DOS disk.
- All Macintoshes can read and write to PC formatted floppies without problems as long as the File Exchange system extension is installed and enabled. If it isn't, the Macintosh will report errors that the floppy is unreadable.
- Format all Zip disks using the *Iomega Tools* utility as it allows you to create DOS readable, removable disks. Unlike floppy disks, the Macintosh can't directly format DOS-compatible disks without using special software.
- All Macintoshes can read and write to PC formatted Zip disks without problems as long as the File Exchange system extension is installed and enabled. If it isn't, the Macintosh will report errors that the Zip disk is unreadable.
- Using CD-R mastering software such as Adaptec's *Toast*, burn a CD using the ISO 9660 format with *Joliet* extensions. *Joliet* enables you to create CDs that are readable on DOS, Linux, and Windows machines while preserving the appropriate filename length for each platform. Be sure to limit your filenames to 31 characters, however, as the Mac OS doesn't support filenames that are longer.
- All Macintoshes can read PC and Linux formatted CD-ROMs provided that Foreign File Access, ISO 9660 File Access, and High Sierra File Access are enabled.

Linux Disk Compatibility with Different Platforms

- Use the free *Mtools* suite of utilities or a similar utility. *Mtools* allows Linux users to format PC-compatible floppies as well as read, write, and manipulate DOS files.
- Most versions of Linux can mount IS0-9660 (DOS) and ISO-9660 (Windows 95/98/NT/2000) CD-ROMs.
- There are also various free utilities available that allow Linux users to read and write files to DOS formatted Zip disks as well.

DOS Disk Compatibility with Different Platforms

- DOS can format floppy disks readable by all systems as long as you stay away from extended formats. This means stick to the plain vanilla 1.4 MB disk format created by most DOS formatting utilities.
- DOS can't normally read Macintosh formatted floppy disks, removable disks, or CD-ROMs. However, using a utility such as *MacSee* by REEVEsoft, Inc. will enable DOS users to access most Macintosh formatted media.

Windows Disk Compatibility with Different Platforms

- Windows can format disks readable by all systems as long as you stay away from extended formats. This means stick to the plain vanilla 1.4 MB disk format and don't use extended disk formats, such as DMF, if you plan on exchanging your disks with other platforms.
- Windows can format DOS-compliant Zip disks using the standard *Iomega Tools* that ships with these drives.
- Using software such as Adaptec's *EasyCD Creator,* all versions of Windows other than version 3.1 can burn ISO-9660 (DOS, Linux, and Macintosh readable) and ISO-9660 Joliet format (long filename CDs readable by Linux and Macintosh systems) CDs.
- Windows machines can't normally read Macintosh formatted floppy disks, removable disks, or CD-ROMs. However, using such utilities as *TransMac* by Acute Systems will enable them to access files stored on most types of Macintosh media.

NOTE: Evaluation copies of both *MacSee* and *TransMac* are included on the book's accompanying CD-ROM. Please refer to Appendix B for more information on these programs.

File Compatibility

When it comes to transferring files between different platforms, files fall into two distinct categories: those that require some sort of conversion and those that don't. Let's tackle the ones that don't require conversion first.

File Format Conversion Issues

One of the nice things about the abundance of software in the marketplace is the fact that there have emerged some informal standards in terms of compatibility between graphics applications and the files they create. This section summarizes:

- **BMP files**—The BMP format is common enough that most non-Windows applications will support them as an input format with little or no problems. This is especially true on the Macintosh when *QuickTime* is installed. Both DOS and Linux offer more limited application support for the format but will work with it.
- **GIF files**—GIF files are accepted by virtually all major graphics applications as an input format and their content displays correctly whether you're on DOS, Windows, a Macintosh, or even a Linux box. For the time being, they are your best cross-platform file format choice.
- **JPEG files**—Like GIF files, JPEG files are accepted by virtually all major graphics applications as an input format and their content displays correctly whether you're on DOS, Windows, a Macintosh, or even a Linux box. Just recall the discussion in Chapter 3 on JPEG's limitations before using them extensively.
- **PCX files**—PCX files have been around a long time and have therefore earned a place as the de facto standard on many platforms. DOS applications have the most extensive PCX support and there are an ample number of programs on the Windows platform that are compatible with this format as well. The same does not hold true for the Macintosh and Linux platforms, however.
- **PNG files**—PNG is still a relatively new file format. However, it will soon have the same level of support across all platforms that GIF currently enjoys. Until then, try to be somewhat circumspect regarding its use until all graphics applications and programming tools are updated to be compatible with it.
- **PSD files**—*Photoshop* files generated by one platform can be read by another (i.e., Windows to Macintosh and vice versa) as long as the target machine has a version of *Photoshop* that's compatible with the original file.
- **TIFF files**—TIFF files are very similar between systems like the Macintosh and Windows. However, they're not completely identical. There's a subtle difference in how they're saved that prevents them from being read directly on either platform without taking special precautions. So, if you plan to use TIFF files on other platforms, make sure you save them with the correct byte order.

This is because there's a specific byte order for both Windows and the Macintosh and neither platform is compatible with the other.

Aside from these exceptions, using all other graphic file formats across platforms is a crapshoot. Most of the other formats mentioned in Chapter 3 are either proprietary to a specific piece of software or only have limited support from graphics applications on different platforms.

File Naming Across Platforms

As discussed earlier in this chapter, each platform has different file naming standards. Table 4-3 highlights how you would handle this issue while moving files across different platforms:

TABLE 4-3: An Overview of How to Handle Filenames Across Platforms

Source Platform	*Destination Platform*	*Comments*
Linux	DOS/Windows 3.1	N/A
Linux	Windows 95/98/NT/2000	N/A
Linux	Macintosh	N/A
Macintosh	DOS/Windows 3.1	If you know you'll be moving files over to DOS ahead of time, cover your bases by limiting your filenames to the DOS 8.3 convention. Otherwise, DOS will mangle your filenames. If you're using Macintosh formatted media, a utility such as *MacSee* will allow DOS/Windows 3.1 users to read the Macintosh files and convert them into the proper filename length.
Macintosh	Windows 95/98/NT/2000	These versions of Windows can handle the Macintosh file length but will need a program such as *TransMac* to read Macintosh formatted media. If the source media is PC compatible, then there's no conversion necessary.
Macintosh	Linux	N/A
DOS/Windows 3.1	Windows 95/98/NT/2000	Don't do a thing. These versions of Windows are fully backward compatible with DOS and Windows 3.1.
DOS/Windows 3.1	Macintosh	Don't do a thing. The Macintosh with File Exchange enabled is fully compatible.
DOS/Windows 3.1	Linux	Don't do a thing. With a utility such as *Mtools*, Linux is fully compatible.

Source Platform	*Destination Platform*	*Comments*
Windows 95/98/NT/2000	DOS/Windows 3.1	If you know you'll be moving files over to DOS ahead of time, cover your bases by limiting your filenames to the DOS 8.3 convention. Otherwise, DOS will mangle your filenames.
Windows 95/98/NT/2000	Macintosh	Limit your filenames to 31 characters. Other than that, the Macintosh with File Exchange enabled is fully compatible.
Windows 95/98/NT/2000	Linux	With the proper tools installed on the Linux end, Linux is fully compatible.

Compressed Files

One last issue to consider is that of compressed files. Compressed files are extremely useful for efficiently exchanging files between platforms. Unfortunately, there exist a number of different compression formats, and each platform has its own means of dealing with them.

Table 4-4 provides information on these file compression formats.

TABLE 4-4: Common File Compression Formats

Platform	*File Format(s)*	*File Extension*	*Comments*
Linux	gzip	.gz	Gzip is the standard file compression format used on UNIX type systems. Almost all Zip-compatible compression programs can read and write gzip files.
Macintosh	Stuffit	.sit	Stuffit is the standard Macintosh file compression format. Support for it is scarce on the Windows and DOS platforms. However, Aladdin System's *Stuffit Expander* can read them.
DOS/Windows 3.1	Zip, ARJ, RAR, or LZH	.zip, .arj, .rar, .lzh, .lha	Of the formats listed here, Zip is the most common. The Zip format is a de facto standard and any properly created Zip file should be readable on any platform with the appropriate reader. The ARJ and RAR formats are common and tend to be popular in Europe.

Platform	*File Format(s)*	*File Extension*	*Comments*
Windows 95/98/NT/ 2000	Zip, ARJ, RAR, or LZH	.zip, .arj, .rar, .lzh, .lha	Of the formats listed here, Zip is the most common. The Zip format is a de facto standard and any properly created Zip file should be readable on any platform with the appropriate reader. The ARJ and RAR formats are common and tend to be popular in Europe.

NOTE: Included on the book's accompanying CD-ROM are versions of utilities for both DOS and Windows, which can read and write all of the file compression formats described here.

File Backups

Why back up your files? Well, the answer is simple: computer files exist as bits of electromagnetic data. As such, they are vulnerable to a whole slew of circumstances that can render them totally useless in a blink of an eye. Such events include:

- Accidental file deletions (quite common)
- Hard drive crashes (very common)
- System crashes (extremely common)
- Fire, earthquake, flood, or some other act of God (rare, but hey, they do happen)

Imagine how you would feel if you had worked 100 hours on a particular set of game graphics only to lose everything due to a hard drive crash. I don't think you would be overjoyed about the situation, do you? Because of this, you're only asking for a world of pain if you don't take the precaution and time to back up your files on a regular basis.

Fortunately, implementing a basic file backup strategy or plan isn't very difficult or time-consuming and there are quite a few options available to you. But, before you start backing up your files, ask yourself some important questions about your specific needs, including:

- How valuable are your files? Do they really need to be backed up after all?
- What would be the consequences of losing your files? Will you lose time and/or money as a result? Or would it not be a big deal?
- Could you easily replace them? If so, how long would it take you to do so?
- Do your files change often? If so, how often? Do you need to keep older versions of your files?

- Are your files very large?
- Do your files need to be secured? Do you need to protect them from unauthorized access or duplication?
- Do you need to transport or distribute your backed-up files to alternate locations?
- Once backed up, how important is the immediate access to them?

Your answers will dictate the solution that's right for you. In a nutshell, you can group these file backup solutions into seven categories:

- Importance
- Change
- Device capacity and performance
- Cost
- Portability
- Reliability
- Security

Backup strategies are based on a combination of these categories, and you should develop a backup strategy that regularly takes these issues into consideration.

Importance

Your files are important to you, aren't they? After all, during the average game project, you'll probably wind up spending hours, days, or even weeks creating and tweaking them to get them looking just right. This being said, you probably wouldn't want anything to happen to them, right? Therefore, any backup strategy you adopt should take into account the potential cost to you in terms of the time and money that you'd need to invest should you need to replace lost files. This in itself should encourage you to back up your files at least every day.

Change

How often your files change figures quite prominently in any backup strategy. Files that change frequently are more prone to corruption and loss than files that do not. To make matters more complicated, these files always seem to be your most important ones! Your strategy should include a plan on how to handle them, such as backing up files that change frequently to ensure that a recent copy of all files exists at all times.

Device Capacity and Performance

Different backup technologies offer different levels of storage capacity and performance. Some systems are capable of copying and storing many files very quickly while others are only capable of copying a few files relatively slowly. Furthermore, some backup devices might not be available for your particular system. Therefore, your backup strategy must consider your particular hardware setup and operating system, as well as personal needs. For example, you may find that certain backup technologies aren't available for your computer. In addition, you may discover that your particular need for backup files is very limited and you don't have to resort to using a sophisticated backup solution.

Cost

Your backup strategy must weigh your financial situation with that of the importance of your files. Faster, higher capacity, and more full-featured backup devices tend to be much more expensive than slower, more limited backup devices. Also, your time is valuable and making backups takes time. You need to evaluate whether your files are worth the cost of what you could be doing with your time in lieu of making backups.

Portability

The portability of different file backup media and devices can also influence the strategy you choose to implement. For instance, in situations where files must be circulated within your office or sent to another location, you will probably want to use a backup device to physically transport your media. You should also choose a backup device that uses media which is compatible with other devices commonly found and used in the environments you regularly access.

Reliability

Different backup devices also offer different levels of reliability. Some backup media is more prone to sporadic corruption than other media. You should take this into account before committing to a solution in order to avoid any nasty surprises. After all, you don't want to go through all of the trouble of backing up your files only to find them ruined by backup media that has gone bad, do you?

Security

You should consider the security of your files to be of paramount importance in implementing any backup strategy. The whole point of backing up your files in the first place is to keep them safe. This means that your backups should be protected

against traumatic events such as theft, fire, and the like. Along the same lines, your backups should also be secure from prying eyes and unauthorized use and access. The last thing you want is for someone to steal your intellectual property. Implementing proper security in your file backups can go a long way in preventing this from happening.

Backup Media

This section will describe some of the more common file backup technologies in common use.

- Floppy disk drives
- Zip drives
- CD-R/CD-RW drive
- Internet-based backup systems

Floppy Disk Drives

Contrary to popular opinion, there's still plenty of life left in the venerable floppy disk and they offer a number of advantages when it comes to backing up small files, including:

- **They're sturdy**—Despite their name, floppy disks are fairly rigid and able to stand significant abuse before failing. This keeps your files reasonably safe.
- **They're cheap**—Floppy disks only cost pennies when purchased in bulk, making them the most inexpensive of all of your backup options.
- **They're portable**—The 3.5" floppy disk is small enough to fit in one's shirt pocket. This makes it easy to transport files using the "sneaker net" across the office or across one's home.
- **They're compatible**—PC formatted floppy disks are readable and compatible with DOS, Windows, and Macintosh platforms. However, the same is not true the other way around. Macintosh formatted floppies are not compatible with DOS or Windows systems without using special software.
- **They're ubiquitous**—With the exception of newer Macintosh models such as the iMac and G3/G4 series, 3.5" floppy disks are installed in and compatible with virtually every PC manufactured since the early 1990s. In fact, there are probably hundreds of millions of installed floppy drives, which virtually assure its ubiquity. This alone makes them a good, inexpensive choice for occasional backup duty.

These points being made, floppy disks have their share of faults, including:

- **They have limited storage capacity**—With average capacities in the neighborhood of 1.4 MB, they only have enough space to handle one or two

640x480, 24-bit images or about eight to ten, RLE compressed 640x480, 8-bit images. This fact keeps them relegated to all but the smallest backup jobs.

- **They're slow**—Floppy drives are notoriously slow, usually offering only about 64 KB/sec of sustained transfer rate in even the best of conditions. Frequently, the host system's OS slows floppy access even more. This makes them unsuitable for fast, frequent file backups.
- **They offer poor security**—Floppy disks are prone to theft and the files contained on them are easily copied. This makes them a poor choice if data security is important to you.

NOTE: I have decided not to mention the so-called LS-120 "Super Disk" technology in this discussion because it is not a widely popular storage format at this time.

Zip Drives

Iomega's Zip disk drive is known as the "super floppy" for good reason. This versatile storage medium offers many advantages, including:

- **Large installed base**—Since its introduction in 1994, the Zip drive is second only to the floppy disk in terms of installed numbers. There are tens of millions of Zip-compatible drives and several times that number of Zip disks in use around the world. These numbers ensure that you're not going to back up your files to a dead medium.
- **Respectable storage capacity**—Unlike floppy disks, Zip disks can store approximately 100 MB of data, which is equivalent to over 70 floppy disks! What's more, newer Zip models support disks that can store as much as 250 MB of data. Even at the 100 MB capacity, the average Zip disk can store a dozen or more full-color 24-bit files or hundreds of RLE compressed, 8-bit graphics files. This is enough for most purposes.
- **Cross-platform compatibility**—Zip drives and media can be used on Windows, Macintosh, and Linux machines. Macintosh systems can read both natively formatted Zip disks and PC formatted Zip disks. However, like floppy disks, Windows machines can't read Macintosh formatted Zip disks without using special software such as *TransMac*. Despite this, Zip disks are very cross-platform friendly.

As with their floppy cousins, Zip drives aren't perfect either. Their problems include such things as:

- **Reliability problems**—Although I've personally never experienced problems with them, there are plenty of users who have. There are some known quality control issues with certain Zip drives that can cause one to either damage or lose their data in certain situations. Therefore, if the integrity of your data is

very important to you, you might want to look elsewhere for a more robust backup solution.

- **Performance issues**—Zip drives come in many flavors, including SCSI, USB, and parallel compatible models. Zip drives operate similar to floppy drives and aren't speed demons to begin with. How you interface the Zip drive to your system will have a significant influence on how fast a Zip drive will copy your files. In general, SCSI-based Zip drives are the fastest, followed closely by USB drives, and, finally, parallel drives. If backup speed is an issue with you and you have a slow Zip drive, it may not be the backup solution for you.
- **Expensive media**—Zip media is significantly more expensive than floppy disk media or even CD-R media. Prices have not gone down substantially over the years, making the cost per megabyte on the high side of available backup storage options. This is certainly something to consider, especially if you're price conscious.
- **Poor security**—As with floppy disks, Zip disks are prone to theft and the files contained on them are also easily copied. This makes them a poor choice if data security is important to you.

CD-R/CD-RW Drives

CD-R/CD-RW drives are recordable versions of the compact disc. At one time, CD-R (compact disc-recordable) and CD-RW (compact disc-rewritable) drives were out of reach for the average person. However, with the massive price drops of recent years, this is no longer the case. Compared to floppy or Zip disks, CD technology offers some incredible advantages such as:

- **Massive storage capacity**—Even in today's world of multi-gigabyte hard drives, the CD's 650 megabyte capacity is still nothing to scoff at. You can potentially store hundreds of full-color, 24-bit, 640x480 images on a CD or thousands of RLE compressed, 8-bit color, 640x480 screens. There's plenty of room for either and then some.
- **Reliability**—Compact discs store data digitally and not magnetically. Therefore, unlike magnetic media such as floppy disks or Zip disks, CDs aren't prone to the same hazards. This means files stored on CDs will usually last much longer than files stored on magnetic media. In fact, this data can often last for years, rather than weeks or months, without any noticeable problems or degradation.
- **Inexpensive media**—CD-R media ranks just next to floppy disks in terms of price. Simply put, CD-R media is very inexpensive. This makes it a good choice for making numerous backups of important files. The same does not hold true for CD-RW media, however, as it's still on the high side.
- **High degree of portability**—CD-R media (not CD-RW) can be read in practically any CD-ROM drive currently in use as long as they are formatted in the

ISO-9660 standard. As virtually all systems manufactured since the early 1990s come with CD-ROM drives, this makes CD-R media almost as ubiquitous as the floppy disk.

Of course, CD-R and CD-RW drives do have their share of disadvantages, including:

- **Write once medium**—CD-R drives can only write contents onto a CD one time. After that time, the disc may no longer be recorded on. While CD-RW drives do offer the ability to write to discs multiple times, they generally can't write to discs at a file by file level. This makes them useless for backing up frequently changed files.
- **Slow performance**—Most CD-ROM drives operate at a maximum of only 300-600 KB/sec. While much faster than either floppy disk or Zip drives, this speed still pales in comparison to the average hard drive. What's worse, since most CD-R and CD-RW drives don't yet operate at the packet level, you must write the contents of the whole CD at once. And, if you're backing up several hundred megabytes of files, this can turn into a quite lengthy process.
- **Limited compatibility**—CD-R and CD-RW media is incompatible with many older CD-ROM drives. What's more, CD-RW discs aren't as cross-platform friendly as, say, the average floppy or Zip disk is. If a CD-R isn't written using the cross-platform ISO-9660 format, one could encounter compatibility problems between DOS, Windows, and Macintosh systems. Again, most Macintosh systems can usually read the contents of PC formatted CDs but the same is not true the other way around.
- **Poor security**—CD-R and CR-RW media is difficult to copy-protect. This means that their contents are easily open to unauthorized access. This makes them a poor choice if data security is important to you.

NOTE: Contrary to popular belief, CD-R/CD-RW media does not last forever. The longevity of recordable CDs depends strictly on the quality of the media used. Therefore, I strongly recommend perusing the official CD-R FAQ pages at `http://www.fadden.com/cdrfaq/` for more information.

Internet-Based Backup Systems

The concept of an Internet-based file backup system is a relatively new and exciting one. It offers users such advantages as:

- **Strong security**—Unlike any of the other backup devices mentioned so far, files that are backed up via the Internet are less prone to physical damage, i.e., theft, fire, etc., because they are actually stored at one or more remote facilities. This makes Internet file backups an interesting and potentially more secure alternative then some of the other devices discussed here.

- **Encryption**—Most Internet-based file backup systems can take advantage of existing and established Internet-based security and encryption protocols. This gives your files an extra measure of security by keeping them safe from unauthorized access or use. On the other hand, most floppy, Zip, and CD-R/CD-RW devices can't provide this without special software or tricks.
- **Convenience**—Anyone with an Internet connection and a Web browser can use an Internet backup system. Furthermore, you can access such backup systems from school, work, or even the comfort of your home. If you need to back up files frequently, this could be a viable solution for you.
- **Universal compatibility**—Because they run off the Internet, such backup schemes don't require expensive hardware or complex software. You can use such systems independent of the platform that originally created the files.

Like the other backup technologies mentioned here, Internet-based backup systems have a number of disadvantages, including:

- **Slow file transfer speed**—Unless you have a very fast Internet connection such as a T-1, DSL, or cable modem, Internet-based file backups can be tedious and slow. Because of this, you might not want to use this as your primary backup system, especially if speed is important to you.
- **Limited storage capacity**—Internet backup schemes tend to be limited in storage capacity. Although different services vary, most seem to limit you to between 10 and 100 megabytes of file storage. In comparison, CDs and Zip disks offer this and more.
- **Cost**—Some Internet file backup systems are free; however, most of them cost between $30 and $50 per month. So, unless you have specific needs, you may find other backup technologies cheaper in the longer run, especially if you're on a budget.

NOTE: I have had good experiences with both FreeDrive (`htttp://www.freedrive.com`) and iBackup (`http://www.ibackup.org`). Accounts for these Internet backup services are either free or low-cost and you can store between 50 and 200 megabytes of information per account. That's quite a deal considering the extra security and peace of mind these services provide.

NOTE: I decided not to mention tape backup devices in this discussion because tape is largely a sequential backup medium. In other words, with tape you will usually be forced to back up everything at once rather than just a few files. Tape drives are also not very common in most homes and small office environments, which makes them less attractive, especially if exchanging files with other people is important to you.

File Backup Strategies

Everyone will have their own tastes and preferences when it comes to backing up their files. And, that's fine. I don't want to suggest something that will cramp anyone's style. However, if you consider your files important, you will adapt some sort of backup system right away.

To minimize data loss and computer downtime when a hard disk crash or computer failure occurs, you should follow these rules when backing up your data:

- **Daily Basis**

Without fail, back up your important files on a daily basis. As files change constantly, the loss of even one day's worth of files can be devastating.

Perform a backup of all of the files and directories that have changed since you last worked on them. This can be done manually, but please be aware that modern operating systems and graphics applications tend to modify many files simultaneously. Therefore, you can't always be assured that you'll have accurate file backups if you perform the backup procedure manually. Instead, consider investing in software that performs what is known as an *incremental backup*. Incremental backups select all the files that have changed since the previous backups by consulting the archive date in a file's directory entry. When the backup is completed, the archive information is then updated to indicate which files have been backed up. Doing this allows files to be ignored the next time a backup is performed unless they have been modified or changed, thus saving you time.

I also suggest backing up your files immediately before turning off your computer for the day/night. Just be aware that this can take a while.

NOTE: Some backup programs can automatically back up your files based on a predetermined schedule. Be sure to look into it as this can be a real time saver.

- **Monthly Basis**

As your time allows, try to back up your entire system at least once a month. This should be in addition to your daily file backup regimen.

Perform a full backup of your hard disk. Again, this can be done manually, but isn't recommended. You're better off using specialized backup software that will ensure that all files are properly copied. The backup software will also ensure that future backups go faster and more efficiently since it will be able to track the history of changes to your files and drives. Once you have successfully backed up the contents of your hard drive, place these backups in a safe place.

Weekends or lunchtime are particularly good times to perform this fun task, as a full system backup can take anywhere from 30 minutes to several hours depending on the size of your hard drive and number of files involved.

NOTE: You are encouraged to make frequent backups of your backups! Like anything else, even backups can go bad. Plan for this and you'll thank me later, especially after you experience your first system crash and attempt to restore your files with a corrupted backup copy. To keep your backups safe, always store them in a cool, dry, and static free environment. This is especially true for magnetic media, as the information on them tends to expand when exposed to hot temperatures. Static electricity can also blank most magnetic media in seconds, so keep your backups away from anything that can accidentally generate a static charge as well.

A Final Word about File Backups

Try to maintain at least two sets of backups made using the incremental backup procedure and rotate these sets in order to be prepared for a system crash or a similar computer disaster. Doing this ensures that your file backups are always stable and fresh in case the worst should happen.

For purposes of illustration, here's a rundown of my own backup strategy:

- I save all of my files every 10-15 minutes. This minimizes the possibility of losing any changes if my machine crashes or the power suddenly cuts out.
- At the end of every work session, meaning at least once a day, I back up my current work files to a fresh Zip disk.
- Before I go to bed, I always make a Zip-compressed archive of my current work files and upload them to one of my `http://www.ibackup.org` or `http://www.freedrive.com` Internet storage accounts. Doing this gives me an extra level of security as it provides me with a second set of backups that is off premises in case my Zip disk becomes corrupted or damaged. Furthermore, I can use this system as a means of archiving and tracking older versions of my files in case I ruin my originals.
- Every two weeks I make backups of my Zip disk to ensure that I always have a good backup copy to restore from.
- Once I complete a major project, I burn a CD-R using high quality media. This gives me a reliable and largely indestructible master set of files.

NOTE: To get your backup strategy off to a good start, I've made arrangements to include the full-featured shareware version of *Fileback PC*, a powerful, low-cost backup utility, on the book's accompanying CD-ROM. Refer to Appendix B for additional details on this program.

Version Control

Version control is a system by which you can record and track the history of your files. This allows you to go back to an older version of your file at any time, and to monitor what was changed, when it was changed, and by whom.

Traditionally, version control has been most at home with programmers who need to keep track of constantly changing source files. However, everyone involved with managing ever-changing files can benefit, especially those who design computer game artwork.

How Version Control Can Help

You can use a version control system to:

- Go back in time
- Compare revisions
- Preserve content safely
- Lock files

Go Back in Time

With a version control system, you can go back in time and retrieve virtually any version of your graphic files at any time. Under version control, every change to a file is saved and marked as a revision. Every revision has unique identifying information saved with it, allowing you to restore files that have changed two, three, ten, or even a hundred times. This allows you to restore "good" versions of your files in case you recently made changes with them and weren't too pleased with the results.

Compare Revisions

Using a version control system, you can easily see what was changed between any two versions of a given file. This is very useful for studying how your files have changed over time, for better or for worse.

Preserve Content Safely

A version control system allows you to preserve the content of any file you create. Your work is never lost no matter how badly you mangled an image because a pristine, or "safe," version of it is always available. This alone makes such a system useful given the frequency that changes and, ultimately, mistakes are made to game artwork.

Lock Files

In a multi-user environment, a version control system can also lock files and prevent two users from modifying (and ruining) the same file. You can also keep tabs on who makes changes to your files and when. This lets you take control of your files and prevents them from being damaged accidentally by other people.

Granted, using a version control system isn't for everyone and they do have their disadvantages, including:

- **Complexity**—Most version control systems tend to be very difficult to learn due to their overall complexity and wealth of esoteric features and options. This in itself can turn potential users off to such a system. To make matters worse, very few version control systems run under Windows, making them even more difficult to operate and configure.
- **They're cumbersome to use**—Similarly, version control systems can be difficult to use. Some people won't like them because they place too many restrictions on their workflow, while others won't like them because they slow down or interfere with the creative process.
- **They're pricey**—Some version control systems can be expensive to buy and implement. That generally makes them unattractive to folks on a tight budget.

Implementing Version Control

Implementing version control can be as simple as installing a version control system on your computer and checking your files into it. However, it's also possible to use a "poor man's" implementation as well, especially if you don't want to go to the trouble of working with such a system. To do this, simply append a version number at the end of your files. For example, to distinguish between version 1 and version 2 of the file `sky.bmp`, you'd name the file `sky.bmp.1` and `sky.bmp.2`, etc. This technique will work fine for Macintosh systems and all versions of Windows except 3.1. It also won't work for DOS as it can cause some applications to choke since they're expecting a standard, three-character file extension, not to mention an eight-character filename. Still, this system will work as long as you adhere to it.

A Final Word about Version Control Systems

For small projects involving few files, I probably wouldn't recommend employing a version control system. It's just too much trouble and fuss. However, as your projects get larger, more complex, and start to involve more people, you may want to use one. Version control systems work best in these scenarios, as their revision tracking and file locking features really go a long way in avoiding catastrophe.

NOTE: To help introduce you to version control, I've included a copy of *CS-RCS*, a full-featured Windows based version control system on the book's accompanying CD-ROM. Check Appendix B for more information.

Basic Asset Management

What exactly is *asset management*? In a nutshell, asset management at its simplest level is a process by which one can organize their graphic images. As you begin taking on larger and larger game projects, it will quickly become evident that your graphic files will soon become hard to manage. Even the most well organized person may eventually encounter difficulty finding specific images as the images accumulate. Just as one would employ a filing system for their important documents, one also needs a filing system for their collection of graphic files. This is where a dedicated asset management system comes in. With one, you'll be able to find even your most obscure files quickly and easily.

The advantages of using a dedicated asset management system versus simply lumping all of your files together are many. First, with a dedicated asset management system, you can generate a database of all of your images. This makes your files easier to manage and track as you add or remove files from your overall image collection. Second, an asset management system will allow you to associate comments and notes with each image. This allows you to provide details on each file in your collection. For example, you could make a note to yourself about a specific file's use in a game or special issues about its palette. Third, a dedicated asset management system will allow you to generate thumbnails of all of your images so you can easily visually browse through your image collection to make sure that you find the file you're interested in. In addition, asset management systems can record other crucial information about your files such as their creation date, modification date, and physical dimensions. You can't even come close to doing any of these things using directory naming schemes and frankly, I wouldn't even try.

Do You Really Need an Asset Management System?

Okay, an asset management system sounds great but do you really need one? Well, that really depends on your particular situation. If you're confident that your game projects will seldom grow beyond a handful of files, say, a dozen or so, then no, you probably don't. For that small amount of files an asset management system would just be overkill and probably doesn't make sense. However, if you start seeing a trend where your game projects routinely generate dozens or even hundreds of graphic files, an asset management system may be just what the doctor ordered.

Choosing an Asset Management System

As one can expect, asset management systems come in all shapes and sizes as well as price ranges. The most sophisticated implementations utilize network based client-server architecture and can easily handle several hundreds of thousands of assets. As our needs tend to be a bit more modest that that, we can get away with a much more rudimentary system.

For our purposes, I've identified several features that an asset management system should have:

- Database driven
- Thumbnail support
- Asset keywords
- Large catalog capacity
- Multiple image catalogs
- Low cost

Database Driven

The asset management system should support a method of adding all of your files into a database. This makes tracking and managing your graphic files a breeze, especially once your image collection starts to contract and expand. Don't use any asset management system that doesn't offer this feature because it's so useful to your asset management strategy.

Thumbnail Support

The asset manager should allow you to browse the contents of your image collection via thumbnails, or smaller versions of your original images. Thumbnails can make it easy to quickly pick out specific images from your collection as opposed to manually hunting for files. Don't use any asset management system that doesn't offer this feature because it can be such a time saver.

Asset Keywords

The asset manager should allow you to add notes, comments, keywords, or any other written data to your image collection. This feature enables you to quickly search for images that meet one or more criteria. For example, you could use it to quickly locate all images in your collection that contain the phrase "spaceships" and that are 320x200 pixels in size. Although this isn't required to make your asset management system useful, it's definitely a nice feature to have and is guaranteed to save you countless time on larger game projects.

Large Catalog Capacity

The asset manager should be able to handle image catalogs of at least 1,000 distinct files. This should give you more than enough room to manage assets for even the largest of game projects. This isn't a requirement, but again, it's another feature that's nice to have. After all, if your asset manager runs out of room in its database, the whole point of using an asset management system in the first place becomes moot, right?

Multiple Image Catalogs

Any asset management system you employ should be able to work with separate image databases for each of your game projects. As you take on more projects, you'll find this feature handy as you try to leverage images you created for older projects for use with your new ones, etc. Such a feature also makes managing your assets that much easier.

Low Cost

This goes without saying! Although asset management is important, it shouldn't be too expensive. Many of us are on budgets and don't need the "firepower" offered by many of the high-end asset management solutions. Look for something that meets or comes close to meeting most of the features here for the lowest price and you'll be in good shape.

One product that comes to mind is *Extensis Portfolio*. I like *Portfolio* because in addition to providing all of the features mentioned here and more, it's cross-platform (runs on Windows 95, 98, and NT 4.0, and the Macintosh). It also costs less than $100.

NOTE: For your convenience, there is a 30-day trial version of this program on the book's accompanying CD-ROM. Please refer to Appendix B for more information.