CHAPTER 21

Memory Management

In This Chapter

Understanding and controlling memory usage

Preloading cast members

Setting purge priorities

Designing movies with memory usage in mind

Outputting to CD-ROM

Director is an enormously flexible authoring tool. It enables you to create movies for a broad variety of uses. With Director, you can design movies for playback on a variety of computers or other output devices capable of digital display. Certainly, you need a powerful computer to produce professional-quality Director movies. Remember, however, that there may be a vast difference between the computer that you use to create Director movies and the one that the end-user uses to view them.

A critical element in designing a movie is a clear assessment of your audience and the capabilities of their computers. In most cases, it is a serious mistake to presume that the end-user's computer is equal to the machine that you use to author your Director movies. Because of those potential differences, memory management is more than just good sense; it's essential for the success of your movies. For a movie to be successful, you need to design it for the viewer. In this chapter, you explore memory considerations both for authoring and for the end-user.

Understanding Memory

Casual computer users frequently confuse memory with hard disk space. Part of this misunderstanding arises from the fact that many computer programs use virtual memory created from hard disk space to supplement physical memory, or RAM. Adding the concept of video memory can increase the confusion. All three of these types of memory are considered part of a computer's resources and are interrelated.

For any discussion of resources to be helpful, you must have an understanding of the different types of memory and, in the case of Director, their combined impact when authoring or playing a movie:

- **RAM:** RAM (random-access memory) is hardware installed on your computer. Depending on the computer you use, these memory chipsets can be hard-socketed (permanently soldered) to your motherboard, or installable (plugged into special slots) on your motherboard. Some computers have a base amount of RAM hard-socketed on the board, plus sockets for expanding the amount of physical memory. How fast your computer operates and its power are determined by the speed of the CPU (central processing unit) and the amount of physical memory, or RAM.
- **Virtual Memory:** Virtual memory is pseudomemory. You create virtual memory by allotting a portion of your hard disk to serve as temporary auxiliary memory for software. Unlike RAM, virtual memory does not affect the base speed of your computer. Rather, programs use virtual memory to increase the amount of memory available. Virtual memory is slower than RAM because your computer's CPU must read and write to use the virtual memory.
- **Video Memory:** Video memory is also hardware-related memory. This memory is installed on your video display adapter, and its configuration can vary from one video board to another. Most video boards today have a minimum of 1MB of memory on the board. Some video cards are expandable, enabling you to add more memory as your needs increase. The more video memory, the more color depth is displayed on your monitor. In addition to controlling the color depth, video memory also affects the screen refresh rate. Performance and faster screen refresh rates increase proportionately with the amount of video memory.

Director works with your computer to use a combination of these resources when you create and play back a movie.

Determining memory requirements

Like most powerful programs, Director requires substantial resources for authoring. The program's basic needs, plus the needs of the movie that you're creating, determine the amount of resources required. Obviously, movies that are more complex require more resources. Director provides the tools to check the requirements of your movie, the available system resources, and information about the various cast members in your movies.

In the following exercise, you learn to use the Message window to check system resources, a movie's memory requirements, and the amount of contiguous memory available. *Contiguous memory* is memory all in one chunk, the opposite of fragmented memory. For some tasks, a contiguous chunk of memory is needed.

Preventing auto-syncing

It's logical to presume that the more virtual memory you have, the better off you are. In other words, if you have a large hard disk and you enable the software to use as much of that space as needed for virtual memory, you might think that your software and computer would perform better. This is not true, however.

Allowing large amounts of virtual memory creates a situation known as *auto-syncing to disk*, which can actually reduce the performance of your computer. When a program uses virtual memory, it stores information temporarily on the hard disk for use by swapping chunks of data to the disk. This disk swapping is also known as auto-syncing. As virtual memory increases, the amount of data being swapped to the hard disk increases, and more chunks are stored to the disk before they are purged from the virtual memory.

Your computer uses RAM to manage virtual memory. The increased amount of disk swapping (auto-syncing) uses more RAM strictly to manage the virtual memory rather than to run software. Consequently, performance decreases as more RAM is dedicated to this use. In some cases, you can actually see and hear the problem. If your hard disk sounds unusually busy or you see that light blinking on more and more often, there is an unusually high amount of activity going on, and an auto-syncing problem probably exists.

This is not to say that you shouldn't use virtual memory—but for best results, you should minimize virtual memory usage. As the amount of RAM in your computer increases, your needs for virtual memory decrease. RAM is inexpensive as of this writing, and it will pay off to load up on it.

You can find the kiosk1.dir file on the companion CD-ROM in the EXERCISE:CH21 (EXERCISE\CH21) folder.

Checking System Resources and Requirements

1. Open kiosk1.dir in Director.
2. Choose Window➪Message to display the Message window.
3. In the Message window, type **put the memorySize** and press Enter. The amount of memory (RAM) installed on your computer is displayed in the window (see Figure 21-1).
4. In the Message window, type **put ramNeeded (1, 2)** and press Enter. The amount of memory required for kiosk1.dir is displayed in the window.
5. In the Message window, type **put the freeBlock** and press Enter. The amount of contiguous free memory is displayed in the window.

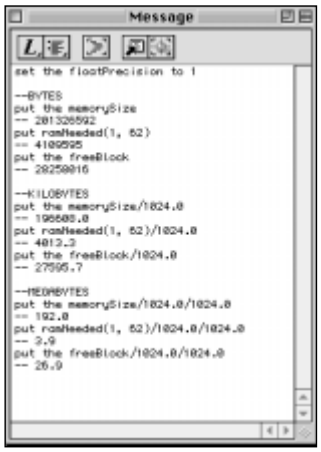


Figure 21-1: The Message window enables you to view memory information.

The memory values are shown in bytes, kilobytes (K), and megabytes (MB). To convert bytes to kilobytes, or to convert kilobytes to megabytes, divide by 1024. Figure 21-1 shows that the computer used for this example has 192MB of RAM. The `floatPrecision` was set to 1 just to make the numbers easier to read (the default is 4). The kiosk1.dir file requires 4,109,595 bytes of RAM, or a little under 4MB.

To load a cast member, you need a `freeblock` at least as large as the cast member. (As mentioned in step 5 of the preceding exercise, a `freeblock` is a contiguous chunk of available memory in bytes.) For example, you could add a Lingo statement to a movie script that identified a `freeblock` large enough to load your largest cast member. If your cast member required 500K, the statement shown in Figure 21-2 would alert the user that memory needed to be freed up to run your movie.

Figure 21-2: You can add statements to a movie script that enable you to check memory and warn users of potential memory problems.

Testing to avoid memory problems

To avoid memory problems, testing is crucial. Most multimedia designers don't have the luxury of knowing the precise hardware configuration of the end-user's computer. The amount of resources, type of video adapters, and speed of the computer vary widely among users.

It is critical that you test your movie during development on a variety of systems. This wise practice prevents unpleasant surprises later. If you are creating a movie for cross-platform use, testing is even more critical. The movie that looks and plays wonderfully on the Macintosh may crash in Windows, or vice versa. Although you can build a lot of safeguards into a movie, nothing replaces testing to ensure the success of your project.

Preloading Cast Members

Preloading large cast members, such as sound and digital video cast members, smoothes the performance of a movie. By default, Director loads cast members as needed. Loading a number of large cast members at the same time can create memory problems. In addition to smoothing the performance of your movie, preloading some of the larger cast members enables you to balance the memory load.

Take care when preloading cast members if you have memory constraints. Preloading too many cast members at one time can cause memory problems.

Lingo can play a significant role in preloading. It can be used to preload cast members, markers, and frames of a movie. Shockwave Audio files (SWA) can be preloaded either in the SWA Cast Member Properties dialog box or by using Lingo. When you use Lingo to preload cast members, you can specify different preload times for each instance of the cast member. This is particularly useful if you need to swap between large cast members that appear more than once in a movie.

Specifying SWA preload values

In the following exercise, you add a Shockwave Audio file (SWA) to kiosk1.dir and you specify the preload time for the file. SWA files offer advantages that are not available with WAV, AIFF, and other sound files. The SWA files are much smaller, and they provide for streaming, which means that they can start playing while they are loading. You can use SWA files in your movies whether they are being produced for the Internet or for running from a local disk. These sound files are linked rather than embedded, so you need to make sure that the SWA file is included when you transport your movies.

It's a good idea to link sound files, regardless of their format, rather than embedding them into your Director movies. Linking sound files enables you to specify preloads and purge properties to conserve memory. Director enables you to convert sound files to SWA format for inclusion in your movies.

For the next exercise, use the kiosk1.dir movie from the companion CD-ROM located in the EXERCISE:CH21 (EXERCISE\CH21) folder. You also use the kiosk1.swa file, which is in the same folder.

Specifying a Preload Setting in the SWA Cast Member Properties Dialog Box

1. Open kiosk1.dir in Director, if it's not already open.
2. Choose Insert ➪ Media Element ➪ Shockwave Audio to display the SWA Cast Member Properties dialog box (see Figure 21-3).

Figure 21-3: Specifying the preload setting

3. Click the Browse button, and select kiosk1.swa from the EXERCISE:CH21 (EXERCISE\CH21) folder.
4. The Preload Time value specifies the number of seconds ahead of the cast member's appearance in the movie that you want the file to start loading. Choose 5 from the list; this is the default setting.
5. Click the Play button to hear the audio. When you're finished, click Stop.
6. Click OK to complete the operation and return to Director's main window. The SWA file appears in the first empty slot of the Cast window. Name the new member **background1**.
7. Save your movie as **kiosk2.dir**.

You can specify SWA file paths by using either a URL or local disk location.

Unlike other sound files, you place SWA files in the sprite channels of the Score rather than in a sound channel. Because Lingo controls SWA files, they can use any of the sound channels during playback, including those not visible in the

Score window. You can specify a sound channel when you insert the SWA file into your movie, or you can let Director use any available channel during playback.

SWA files can be imported as well as inserted. Imported members have the standard capabilities of normal sound members, but they do not have the same capabilities as an SWA that has been inserted. Inserted members act as they have in previous versions of Director. SWA-specific Lingo works only on inserted members. Imported SWA requires MIX Services, SWA Import Export Xtra, and the Sound Import Export Xtra for playback. Using imported SWA members in a Shockwave movie causes all three Xtras to download.

Placing the SWA File in the Score

1. From the Cast window, drag the cast member to frame 1 of channel 22 in the Score.
2. Select the tail frame of the sprite; then click the frame bar at frame 77.
3. Press Command+B (Ctrl+B) to extend the sprite to frame 77 (see Figure 21-4).

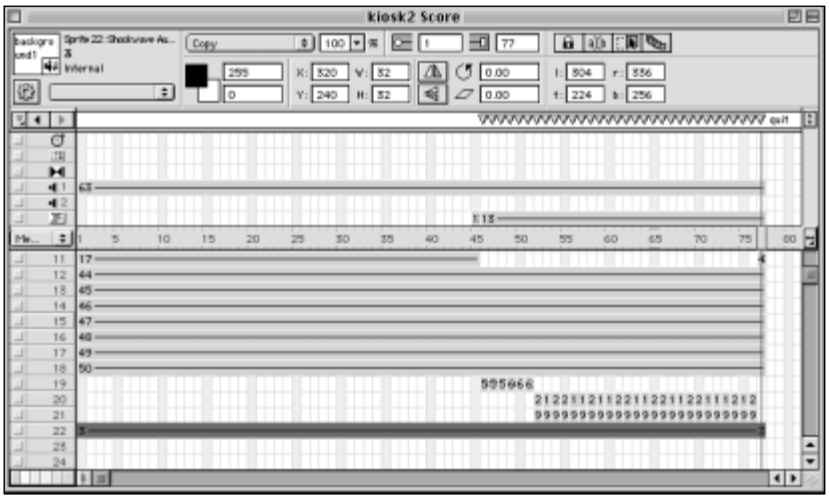


Figure 21-4: The SWA sprite is extended through frame 77 of channel 22.

4. Save the movie again as kiosk2.dir.
5. Rewind the movie and click the Play button to test the movie.

SWA files play on both the Macintosh and in Windows. Using this format offers you the capability to play a movie containing sound on both platforms, even if you don't have utilities to convert the sounds from the native format of one platform to the other. In addition, the small file size of SWAs helps you control the amount of RAM and hard disk space required for the storage of your movie.

Using Lingo to specify preload values

You can use Lingo to specify the preloading of individual cast members, a cast library, your movie, a marker, or a range of frames. This is especially useful if you are working with a movie in a window (MIAW), digital video, or sound cast members. Table 21-1 lists Lingo preload commands and their uses. It also shows an example of the command usage.

Table 21-1
Lingo Preload Commands

Command	*Application*	*Usage Example*
`PreLoad`	Preloads cast members from a range of frames into memory. This is useful for loading cast members that animate. Note that if no parameter is specified, Director continues to load the entire movie, stopping only when memory is full.	`PreLoad marker ("finance")` Loads the marker named `finance` into memory.
`PreLoad 20, 32`	Preloads frames 20 through 32 into memory.	
`PreLoadBuffer`	Preloads an SWA file or portion of an SWA file into memory. Only works if the SWA is stopped.	`PreLoadBuffer (member member "kiosk1")` Preloads the SWA file KIOSK1 into memory.
`PreLoadMember`	Preloads a specific cast member or range of cast members from the cast library into memory.	`PreLoadMember "seq01", "seq38"` Preloads the cast members from seq01 through seq38 into memory.
`PreLoadMode of Castlib`	Specifies when to preload a cast library. Options include the values 0 (when needed), 1 (before frame 1), 2 (after frame 1).	`set the preLoadMode of castLib "sounds" = 1` Preloads the Cast Library named "sounds" into memory before frame 1 of your movie.

Command	*Application*	*Usage Example*
`preLoadMovie`	Preloads the specified movie by preloading its cast members. Prevents delays when the current movie opens another movie or an MIAW.	`PreLoadMovie "menu"` Preloads the movie named menu into memory.
`preLoadNetThing`	Preloads a file from the Internet. This is useful if you have multiple movies on a Web site, by preloading the next movie while the current movie is playing.	`PreLoadNetThing ("http://www.mymovie.com/playit")` Preloads the movie named PLAYIT into memory while the current movie is playing.
`preLoadRAM`	Allocates limited memory for digital video, while enabling other cast members to load normally.	`set the preLoadRAM to 2 (the size of member "cycling")` Specifies that the amount of RAM available for preloading equals twice the size of the digital video cast member named cycling. If `preLoadRAM=False`, all memory available can be used to preload digital video.
`preLoadTime of member`	Specifies the amount of an SWA file to download before it starts playing. The amount is limited by seconds, rather than a number of bytes. Note that the SWA file must be stopped for this command to work.	`set the preLoadTime of member"kiosk1" = 6` Preloads the sound cast member named `kiosk1` 6 seconds before it starts playing.

Setting Purge Priorities

You can set the purge priorities of the cast members in your movies. By default, Director purges cast members at random, as needed to free up memory when it's full. Often it is best to leave it at the default setting and allow Director to handle the memory. If the size of the movie exceeds the available memory, setting purge priorities enables you to control which cast members are purged from memory and which are retained. When Director needs to reload a cast member frequently, it puts a strain on the resources of the user's computer. For example, a bitmap used as a

background throughout a movie should be retained in memory for the entire movie, to prevent repeated pauses for reloading the image.

To specify purge priorities, you specify a value that indicates when a cast member should be removed. The higher the priority value, the more likely it is that a cast member will be purged if memory runs low. The following values are available for purge priority settings:

0 = Never Purge

1 = Purge Last

2 = Purge Next

3 = Purge Normal

For the next exercise, use the kiosk2.dir movie from the CD-ROM, located in the EXERCISE:CH21 (EXERCISE\CH21) folder.

Specifying Purge Properties in a Movie Script

1. Open kiosk2.dir in Director, if it's not already open.
2. In the Cast window, double-click the cast member with the `startMovie` handler (if you are using the CD-ROM's version of kiosk2.dir, it is in cast member 4) to open it in the Script window.
3. At the top of the script, enter the `prepareMovie` handler, as shown in Figure 21-5.
4. Close the Script window, and save your movie.

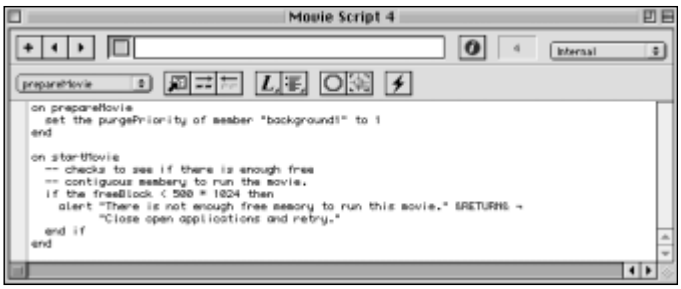


Figure 21-5: Adding purge priorities in a movie script

Director doesn't purge cast members from memory unless it needs the memory to run a movie. On computers with substantial resources, purging cast members is rarely a problem. If you're designing an application for use on a broad range of computers, you should pay close attention to specifying the way cast members are purged from memory, to ensure the success of your movie.

Designing Movies for Memory Considerations

When designing Director movies, it's important to remember your audience. Where memory is concerned, designing for the least common denominator is necessary if you want to reach the broadest audience.

Clearly, if you are designing a project for the World Wide Web, you can anticipate an enormous variety of users and hardware. Because it's impossible to design for every variable, it's a good idea to pick a *target platform*. A target platform is the lowest system your software will run on acceptably.

Guidelines for Saving Memory

Director helps you to control memory usage. In addition, you can cultivate success and improve the performance of your movies by keeping the following considerations in mind when you are designing your movies:

- ✦ Use the lowest possible color depth for bitmapped cast members that have only a few colors. The higher the color depth, the more resources that are needed to render the cast member and refresh the screen.
- ✦ Create composite images of bitmap cast members that appear as static images on the Stage. This reduces the number of cells and channels that Director has to read and process, enabling your movie to run more smoothly.
- ✦ Use the floating Tool Palette to draw and fill simple geometric shapes, instead of employing bitmap images. Objects drawn with tools from the Tool Palette are vector images and require less time and fewer resources to render to the screen. Don't use vector graphics for animation, however, because they will slow your movie enormously.

Bitmap memory considerations

It's a good idea to keep memory requirements in mind when you create bitmap images for your movies. Bitmaps can account for a rather large chunk of your final movie size. And if you add sounds and other effects in addition to large bitmaps, your movie may exceed the resources of the end-user's computer. You can use the following formula to predict the amount of memory required to display a bitmap:

> (the width of the bitmap in pixels) × (the height of the bitmap in pixels) × (the color depth in bits) / 8 = the number of bytes of memory required

For example: a 200-pixel × 200-pixel, 8-bit bitmap would require 40,000 bytes of memory, or (200 × 200 × 8) / 8 = 40,000.

- ✦ Keep the use of gradient fills to a minimum. Gradients require more resources to render to the screen.
- ✦ Try not to use memory-intensive inks with complex or large bitmaps, which can bring a movie to almost a dead stop on most computers, even if they possess significant resources.
- ✦ Use film loops for repetitive animation sequences. A film loop will play as a single cast member in a frame while Director is processing other cast members in the frame.
- ✦ Reduce the sampling rate of sound files, or consider using Shockwave Audio (SWA) files to improve performance.
- ✦ Get the most from small sound bites, such as knocks, beeps, and clicks, by setting them to loop.
- ✦ Link rather than embed your sound files, so that Director doesn't have to load the files until they're needed in your movie. When linking files, be sure that the file is included when you transport your movie from one location to another.
- ✦ Use Lingo preload and purge commands to control the memory usage of your movies. This prevents bottlenecks that slow or halt a movie if too many cast members are loading at the same time.
- ✦ Consider using movies in a window (MIAWs) when your movie contains multiple branches of movies that could be designed to run independently. Just remember that you can't use MIAWs on the Internet.

Observing good memory-conservation practices can increase the performance of your movies and make troubleshooting easier. Frequently, as in the case of using film loops, memory conservation produces a tighter, more compact Score. And that makes it easier for you to find elements within your movies and edit them.

Optimizing for CD-ROM

Director movies are large. It's not uncommon for a single movie to be well over 4MB. If you are creating a project in which multiple movies play in succession and have MIAWs, sounds, and perhaps digital video, the amount of space required for storage can mushroom almost geometrically. CD-ROMs have become the distribution media of choice for most Director projects that aren't being published on the Internet. CD-ROM disks have a number of advantages over floppy disks. They are inexpensive and durable, they can't be damaged by magnetic fields, they hold large amounts of data, and they can be built to play on multiple platforms.

The transfer rate of a CD-ROM drive can have an impact on the performance of a movie. As a general rule of thumb, the transfer rate of a CD-ROM drive is 150KB per second times the speed of the CD-ROM. Thus, a 4X CD-ROM would have a 600KB-per-second

transfer rate. Slower drives (below 4X) may cause a performance problem with your movies. Be sure to test your movie thoroughly before distributing it.

Building a virtual image

The first step in recording a CD-ROM is to build a virtual image. A *virtual image* is a database of pointers telling the mastering software where to find the files that you want to include. When you create the virtual image, you are not actually moving any files. You are telling the mastering software how you want the CD-ROM organized, by setting up directories and allocating files to those directories.

Before recording your CD-ROM, it's a good idea to do a little housekeeping to organize your files. By organizing your files into folders and keeping related files together, you improve the performance of your movies by reducing the amount of time the computer has to spend looking for files. In addition, you also reap the benefits of simplifying the task of creating a virtual image and improving your chances of a successful recording session. When you put the files in order on your hard disk prior to starting, you'll be prompted to remove files that will not be included on the CD-ROM, and will more easily recognize a logical order for your files. Remember: You are dealing with almost 650MB of information.

When it's time to move linked files, such as digital movies, sound files, and external casts, you should open and resave your Director movies to update the location of those linked files.

Although most mastering software permits a directory tree of folders and subfolders eight deep, it's a good idea to limit the folder depth to three or four levels. This reduces the seek and transfer times from your CD-ROM to your computer—yet another way to improve the performance of your movies.

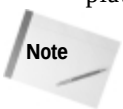
CD-ROM basics

There are two basic recording standards for CD-ROMs: ISO 9660 and Hierarchical File System (HFS). Macintoshes can read both formats, but they prefer the HFS standard; Windows-based machines can read only files that have been encoded by means of the ISO (International Standards Organization) 9660 standard.

It is possible to record both formats on a single CD-ROM. You should be aware, however, that recording your movies in both formats on a single compact disc cuts in half the amount of available storage space. This is the result of including two copies of a movie or projector (one for each platform). If you find that you need to record more than 300MB of data for each format, you'll either have to use some form of data compression or reevaluate your decision to put both formats on the same disk. Not all files need to be duplicated for cross-platform presentations. It is possible to share many of the external files, such as QuickTime movies, via aliases. In fact, QTVR files do not play well from an ISO 9660 volume. In that case, it is better

to store them in a Macintosh HFS volume and reference them through shortcuts on the PC ISO 9660 side of a multiplatform CD-ROM. Because of its audio origin, the CD-ROM blank medium is measured in minutes:seconds:sectors. It comes in two sizes: 63 minutes or 74 minutes. This works out to be 553MB and 650MB, respectively. The 650MB discs are the most common.

It's not a good idea to run too close to the maximum storage amount on a disk. Both the ISO 9660 and HFS standards require approximately 1MB of overhead space for root files, resource forks, and other data required by the operating systems of both platforms.



If you are including data from multiple sources, be certain that you observe copyright laws and restrictions. You must obtain licenses to include any files, including sound and digital movie files. If you are using clip-media sounds, digital movies, text, or artwork, be sure the license rights provide for redistribution of the files.

Single- or multisession recording?

When you record a CD-ROM, you can choose between two modes: single-session or multisession. Single-session only enables you to burn the content onto the CD-ROM one time as a single volume. Multisession CD-ROMs can contain multiple volumes that can be added at any time. When you mount a multisession CD-ROM, it appears on your computer as several different volumes. Single-session CD-ROMs are more useful for mastering a CD-ROM that will be widely distributed. Multisession CD-ROMs, on the other hand, are better for backing up and archiving, because you can add more volumes as the project grows. Your choice depends on the media you choose, the CD-R (CD-ROM recorder), and the recording software.

Understanding logical blocks

Data is stored on a CD-ROM in *logical blocks,* rather than as the total number of bytes of all the files. A logical block is the smallest recordable unit on a CD-ROM. This means that files occupy more space than their original size. Logical blocks can be 512, 1024, 2048, or 4096 bytes in size.

MSCDEX (Microsoft CD Extensions) and HFS can only read a 2048-byte block. Consequently, each file occupies a space equal to the nearest multiple of the next higher block. For example, if a file contains 16,000 bytes, it will require 8 blocks (16,000 / 2048 = 7.812 blocks).

Simply totaling the number of bytes for all the files that you want to record won't yield an accurate estimate of the space required. Most CD-ROM mastering software calculates the amount of space required for your files, but you should already have an estimate of your space requirements *before* you start recording a CD-ROM. Remember that directories, like files, require space on the CD-ROM. Generally, you can record up to 600MB of raw data on a CD-ROM. Under most circumstances, this amount leaves ample space for overhead while taking logical locks into consideration.

Two types of CD-ROM media are available today: CD-ROM (Mode 1) and CD-ROM XA (Mode 2). The difference between the two types is frequently misunderstood — due in part to the history of compact disc media. Early CD-ROMs were recorded in a single session, even though the standard made provisions for recording in multiple sessions. Kodak's Photo CD medium was one of the first to make use of multisession recording, but it did not use the standard CD-ROM format. These early multisession disks used the CD-ROM XA format, so the manufacturers of CD-ROM drives assumed that all multisession CD-ROM media would also be CD-ROM XA format. The reality is that both formats can be used for multisession recording.

Because of the misconceptions arising from the two CD-ROM media formats, you cannot assume that a CD-ROM recorded in multiple sessions will be read correctly. If you anticipate recording in more than one session, you would do well to either premaster the CD-ROM on a hard drive, or make sure that data is recorded in the CD-ROM XA format.

The choice of whether to record in a single or multisession format requires taking a close look at your needs. Each recording format offers distinct advantages and disadvantages. To begin, remember that when multiple people are involved in the production of a single movie project, sometimes it's difficult to gather all the data into a single location simultaneously. If you want to record a single-session CD-ROM, it's essential that the virtual image contains all of the files, organized precisely as they will be placed on the CD-ROM. If you are working on a network, or are including data from remote sources, a single recording session might not be practical.

Multisession recording offers you the advantage of being able to add data at a later time, from a variety of sources or locations. If you are creating a CD-ROM for multiple platforms, it's necessary to create a multisession CD-ROM to accommodate the differences in the operating systems of the two platforms. On the other hand, multisession recording requires more headroom on the CD-ROM, which means it cannot hold as much data.

Summary

Employing memory management techniques helps trim the size of your movies while maintaining a high quality of performance for your projects. As your movies become larger and more complex, good memory management makes a noticeable difference in movie performance. In this chapter, you learned how to:

- ✦ Use SWA files in your movies and to specify the preload times to ensure that they play smoothly.
- ✦ Calculate the amount of memory required for bitmap images.
- ✦ Set purge priorities.
- ✦ Use the Message window to obtain information about memory and your movie.

- ✦ Design movies with memory considerations in mind.
- ✦ Organize your projects for CD-ROM output.

In the next chapter, we discuss Lingo related to user-interface elements, such as dialog boxes and menus.

✦ ✦ ✦