

**M** MACROMEDIA®

Tips  
& Tricks

**Director**®

Version 4

## **Trademarks**

Action!, Authorware Professional, Macromedia, Macromedia Director, Macromedia Three-D, MacroModel, SoundEdit Pro, and SoundEdit are trademarks of Macromedia, and Authorware and MacroMind are registered trademarks of Macromedia, Inc. All other trademarks are the property of their respective owners.

## **Apple Disclaimer**

**APPLE COMPUTER, INC. MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING THE ENCLOSED COMPUTER SOFTWARE PACKAGE, ITS MERCHANTABILITY OR ITS FITNESS FOR ANY PARTICULAR PURPOSE. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED BY SOME STATES. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY PROVIDES YOU WITH SPECIFIC LEGAL RIGHTS. THERE MAY BE OTHER RIGHTS THAT YOU MAY HAVE WHICH VARY FROM STATE TO STATE.**

**Copyright © 1994 Macromedia, Inc. and Apple Computer, Inc. All rights reserved. This manual may not be copied, photocopied, reproduced, translated, or converted to any electronic or machine-readable form in whole or in part without prior written approval of Macromedia, Inc.**

**First Edition: February 1994**

Macromedia, Inc.  
600 Townsend St.  
San Francisco, CA 94103

# Contents

---

<b>Authoring for CD-ROM .....</b>	<b>1</b>
Perceptual techniques.....	1
Managing memory.....	2
Managing sound.....	3
Managing graphics.....	4
Managing animations and QuickTime.....	5
Using a shared cast movie.....	5
Optimizing files for CD-ROM.....	6
Managing Lingo.....	7
Formatting a CD-ROM.....	7
<b>Basic memory-use rules .....</b>	<b>8</b>
Optimizing memory use.....	8
Predicting bitmap size in memory.....	9
Determining how much memory a movie uses.....	10
<b>Improving movie performance .....</b>	<b>11</b>
<b>Synchronizing graphics to sounds and digital video .....</b>	<b>13</b>
Synchronizing with the tempo channel.....	14
Synchronizing with Lingo.....	15
Preloading the cast.....	17
Optimizing for individual machines.....	17
Sound-only QuickTime movies.....	18
Maintaining synchronization in a long animation.....	19
<b>Strategies for debugging Lingo code.....</b>	<b>21</b>
Using the message window.....	22
Using the put command.....	22
Displaying values on the stage.....	24
Script comments.....	24
Using Find/Change.....	25
Common error messages.....	25

<b>Using 32-bit images .....</b>	<b>27</b>
Hardware and system considerations.....	27
Importing 32-bit images .....	27
Gradients.....	28
Using palettes and colors .....	29
Accessing colors not found in existing palettes .....	30
Animating 32-bit images .....	30
<b>Creating movies for Windows .....</b>	<b>31</b>
Ensuring high-quality Windows movies .....	31
Mapping Macintosh fonts to Windows fonts .....	34
Editing a movie's FONTMAP.TXT file .....	35
Using fonts on high-resolution screens .....	41
Managing PC memory.....	42
Getting good color.....	42
Measuring color resolution and color depth.....	43
Playing back in 16 colors.....	43
Playing back in 256 colors.....	44
Using palette effects.....	44
Using sounds.....	45
Using external sounds.....	45
Using sound channels in the score window .....	46
Playing sound on a PC .....	46
Setting sound volume.....	47
Working with different size monitors .....	47
Using QuickTime.....	48
Converting QuickTime movies.....	48
Importing QuickTime movies.....	49
Using Microsoft Video for Windows (AVI).....	49
Naming files and folders .....	50
Choosing filenames and folder names .....	51
Assigning pathnames .....	52
Using relative pathnames in Lingo.....	53
Inks and performance .....	54
Using Lingo commands.....	55
Unsupported Lingo commands.....	55
Partially supported Lingo elements .....	56
Using Lingo properties.....	57
Using MCI .....	58
Sending MCI commands.....	58
Examples of MCI commands .....	59
Working with XObjects.....	60
Using PICT images.....	61
Linking movies .....	62

Creating menus.....	63
Features that behave differently on the PC .....	64
Using stretched sprites .....	64
Controlling external applications .....	64
Transferring files to the PC .....	65
MacLink Plus/PC .....	66
LapLink Mac II .....	66
DOS Mounter .....	66
TOPS .....	66
Apple File Exchange .....	66
PhoneNET PC .....	66
Timbuktu .....	66
Apple PC Exchange .....	66
<b>Frequently asked questions .....</b>	<b>67</b>



---

## *Authoring for CD-ROM*

A CD-ROM drive's performance can be measured by access time (measured in milliseconds), which is the time required to locate a random piece of information; and transfer rate (measured in kilobytes per second (KBps)), which is the time required to transfer information to the CPU. When authoring for CD-ROM, you will usually be more concerned with the transfer rate. The actual data transfer rate of single-speed CD-ROM drives is approximately 90 kilobytes per second (KBps), after factoring in the overhead time required for CPU processing. Faster double-speed drives can transfer data at up to 300 KBps, while transferring audio at 150 KBps. Faster drives are becoming more common.

A good rule of thumb is that in order to play digital video movies, you must use a drive that has at least a 400-ms access time. The faster the CPU and the faster the CD-ROM drive, the more smoothly the movie appears to play back.

### *Perceptual techniques*

- ◆ To give the perception of faster response time, provide instant feedback to users in response to their input, such as highlighting the location where they just clicked. You can then perform some action that may take a few seconds to complete.
- ◆ If there is a delay while Director loads a movie or some other large file, “hide” the load time by:
  - ◇ Displaying a custom “wait” icon such as a watch cursor
  - ◇ Playing an audio message or a sound
  - ◇ Displaying text that the user can read while waiting.

## *Managing memory*

- ◆ Determine the minimum free memory that the movie will be required to use, and then work backward from that minimum to make sure that your movie fits into available memory. For example, you should take into account all global variables and objects, and determine how much memory they will require. After subtracting this amount from the projector size set in the Finder, you will know the amount of memory that remains for both sound and graphic cast members in any one frame.
- ◆ You can check memory usage using the About Director command on the Apple menu. The available memory can also be determined before playing the movie by typing `put the memorySize` in the message window. You can also use `put the freeBlock` to see what's happening to memory while you play the movie. You can also use the `traceLoad` command to display cast member names and other information about cast members as they are loaded into memory.
- ◆ Using Lingo, include a check of the `freeBlock` before the movie starts to determine if the minimum memory necessary to run is available. If not, display an alert informing the user how much more memory is necessary.
- ◆ Use the Cast Member Info command on the Cast menu to tell Director when to purge a cast member from memory. Make sure large cast members have an appropriate purge priority.
- ◆ Load cast members into memory on a “when needed” basis (specified using the Movie Info command on the File menu).
- ◆ If memory on the target machine is a consideration, create separate versions of your Director movie on the CD-ROM, and design each one to run on machines with 5 Megabytes, 8 Megabytes, or 16 Megabytes of physical RAM. Create a separate movie that checks the available RAM and then calls the appropriate movie version.

- ◆ During authoring, use the Lingo properties the `traceLoad` and the `traceLogFile` to determine when cast members are used in your movie. Set the cast purge priority to make sure cast members that are needed frequently don't get purged from memory.

### *Managing sound*

- ◆ Keep sound segments as short as possible by breaking them into short durations. For example, if the movie uses narration, don't put the entire narration into one sound file. Instead, break the narration into smaller files containing individual phrases. This will help manage memory and synchronization. Use the `Wait for sound` setting in the `Set Tempo` dialog box or the Lingo command `soundBusy`.
- ◆ Smaller sound files require less load time when pre-loading a sound into memory. Try down-sampling high-fidelity sounds, as long as sound fidelity isn't adversely affected. The resulting sound files will occupy less disk space. For example, music may have to remain at 22Khz, which corresponds to a transfer rate of 22K per second of storage. However, speech may be acceptable at 11Khz, which corresponds to a transfer rate of 11K per second.
- ◆ If possible, don't link to sounds. Make sure they are embedded in the cast to minimize load time. If you do use linked sounds, store them on a separate disk from your movie files.
- ◆ Preload sounds into memory.

- ◆ Don't redraw graphics while trying to load a sound. Choose natural pauses in the movie when you can perform preloading. For each sequence of frames, preload the sound cast member, which is tweened across these frames, giving it a purge priority of 3, so that it is among the first to be removed from memory. Assign the graphic cast members for the same sequence of frames with a lower purge priority. Play the sequence, and see if graphic cast members can all be loaded into the available memory. If not, you must either increase the memory allocation (Suggested Memory Size) available to Director or the projector or use other techniques to reduce the amount of memory needed by graphic cast members.
- ◆ For playing large sounds, use the `sound PlayFile` command rather than the `puppetSound` command, so that the sound gets read directly from the volume itself, rather than first loading the sound into memory before playing it. Be advised that this technique may interfere with loading cast members while the sound plays, since the volume head cannot be in two places at the same time.
- ◆ Don't play an AIFF sound file from the CD-ROM at the same time you're trying to load a Director file. This will result in much slower load times. Instead, use a short looping puppet sound that plays while the movie is loading.

### *Managing graphics*

- ◆ Keep graphics redraw to a minimum by only animating the part of the image that has changed.
- ◆ Use the trails ink effect to keep unchanging cast members on stage so that they don't have to be redrawn.
- ◆ Use 8-bit graphics instead of 24-bit graphics.
- ◆ If you're authoring for a 68040 machine, try using 1-bit graphics and colorizing them, or else try using JPEG compression for your graphics. (Using JPEG compression on machines slower than a 68040 Mac is not recommended, due to the long load times necessary on slower machines.)

- ◆ Don't stretch sprites, since they greatly reduce performance. Resize the cast member using the Transform Bitmap command on the Cast menu.

### *Managing animations and QuickTime*

- ◆ If you plan to use tightly synched sound and animation at the same time, you should incorporate them into a QuickTime movie.
- ◆ For small rectangular animations, use QuickTime as an alternative to using a Director animation, to avoid having to pre-load many cast members required for a Director animation.
- ◆ If you have a large QuickTime movie, consider breaking it into smaller chunks to avoid having to load the entire movie all at once.
- ◆ If you have a large QuickTime movie, first run it through a QuickTime compression program such as Apple's MovieShop utility. MovieShop adjusts the data rate and quality of a QuickTime movie for optimal playback performance on CD-ROM drives. If the QuickTime movie contains sections where many pixels or colors are changing at once, such as when the movie pans across a scene, the CD-ROM drive may not be able to process the changes fast enough, resulting in slower throughput. Custom compression programs such as MovieShop let you reduce the data rate of the QuickTime movie to match the data rate of the CD-ROM drive.
- ◆ Use the Direct to Stage option in the digital video cast member's Cast Member Info dialog box. This option provide the best possible playback performance from a digital video movie.

### *Using a shared cast movie*

Identify all common elements and place them in a shared cast movie (SHARED.DIR), so these elements don't have to get purged from memory and later loaded back into memory, causing delays. Include global or frequently-used movie scripts in the shared cast.

## *Optimizing files for CD-ROM*

- ◆ By default, files and folders are saved to the CD in alphabetical order. You can override this default organization to improve access time. When you write files and folders to the CD, organize them so that all related elements (Director files, QuickTime movies, and sounds, for example) belonging to the same part of the movie are grouped together on the CD-ROM. This reduces the number of times the playback head has to move around and change speeds to find related files. The files can be all in the same folder, or in separate but adjacent folders.
- ◆ At each decision point, group all files that the user is likely to use next near the movie that user has just finished playing. This strategy reduces access time, since the CD-ROM's read head is already in the general vicinity of the files that the user is likely to use next. This organization can be accomplished by using a file naming convention.
- ◆ Install files that can be randomly accessed on the computer's hard disk, to avoid having to move the CD-ROM's read head to their location each time they are required.
- ◆ Save the movie using the Save and Compact command on the File menu to gain optimum performance for playback on a CD-ROM drive.
- ◆ While you are in the process of creating the movie, play it back on a computer that is slower than the computer you are using to create it. Step through the movie frame by frame and use the control panel to compare the tempo setting with the actual speed the movie is running at. If the difference is too great or performance on the slower computer is too poor, you can make changes to improve playback.

## *Managing Lingo*

- ◆ Keep the number of lines of Lingo code to a minimum to reduce execution time.
- ◆ Sometimes a more round-about way of doing something results in faster execution time. Test the execution time for sections of your code using the `timer`.
- ◆ Avoid using many “if” statements strung together.
- ◆ Long handlers execute more slowly, so use small handlers. Use a separate handler for each action, and use one handler to determine which action is appropriate.
- ◆ Avoid calling a function unnecessarily. If the result of a function is used more than once, save the result in a variable and refer to the variable the next time the function’s result is required.

## *Formatting a CD-ROM*

- ◆ Use a low-level disk format, such as that provided by FWB Toolkit Primer, when clearing a CD-ROM disk for mastering. Low-level formatting improves QuickTime movie playback from CD-ROMs.

---

## *Basic memory-use rules*

This section explains some methods for optimizing memory use. The information describes basic memory-management techniques for the Macintosh and PC platforms, some ways to optimize how Director loads cast members, and how to predict the amount of memory a movie requires on the PC.

Use the same basic memory-management techniques to create a Windows movie that you use to create a Macintosh movie:

- ◆ Be conservative with memory-intensive cast members such as large internal sounds and bitmaps. If possible, use external sounds instead of internal sounds.
- ◆ Keep animations as small as possible. If cast members are too big, you get a memory error message when the movie is played back.
- ◆ Create movies in several smaller segments instead of one large movie whenever practical. Smaller segments are more likely to play back without pausing, and can begin and end at logical breaks in the movie.
- ◆ Try to use text and QuickDraw cast members instead of bitmapped text and shapes. Remember though that QuickDraw cast members animate more slowly than bitmaps. Use your judgment about whether memory use or animation speed is more important.
- ◆ Use trails to leave sprite images on stage if the sprite doesn't move and uses no transitions.

---

## *Optimizing memory use*

When the total size of cast members is larger than the available memory, Director unloads cast members and reclaims their memory as the movie plays. Cast members that have gone the longest without being used are freed from memory first. This reduces the chance that Director will need to pause to reload a cast member.

Loading cast members as they are needed reduces the demands on memory.

- ▶ To have cast members load only when they are needed, choose When Needed in the Load Cast option in Director's Movie Info dialog box.

Use the Cast Member Info command on the Cast menu to tell Director when to purge a cast member from memory. Make sure large cast members have an appropriate purge priority.

- ▶ **Tip** *The movie pauses each time a cast member is loaded. To avoid numerous short pauses to load each cast member, use the Lingo commands `preLoad` and `preLoadCast` to load several cast members at a time.*

---

## *Predicting bitmap size in memory*

Most cast members are the same size in memory as they are on disk. However, bitmaps can change size when they are loaded into memory, depending on the color resolution of the playback monitor.

A movie's color depth is determined by the color depth of the monitor when the movie was saved. A cast member's color depth is determined by the monitor's color depth at the time the cast member was created or imported.

A 1-bit color depth image remains a 1-bit color depth image, but 4-bit and 8-bit images adjust their color depth depending on which computer they play on. This has two implications:

- ◆ A 4-bit cast member expands to twice its original file size in memory on a 256-color monitor.
- ◆ An 8-bit cast member shrinks to half its original file size in memory on a 16-color monitor.

The following formula measures how much memory a bitmap uses:

*Number of bytes in memory used by a bitmap=(image's height in number of pixels x image's width in number of pixels x color depth in bits)/8*

For example, a bitmap image that is 100x100 pixels and has a 4-bit color depth uses 5,000 bytes of memory (100x100x4)/8=5,000).

---

## *Determining how much memory a movie uses*

To see how much memory a movie uses on a Macintosh:

1. Open the movie on the Macintosh.
2. Choose the About Director command from the Apple menu.  
The bar graph shows how much memory the movie requires.

To determine how much memory a frame of a movie uses, use the `ramNeeded` Lingo function.

► **Tip** *The following Lingo script displays the size of an individual cast member in the message window:*

```
put the size of cast name-or-number
```

*When you use this command, replace “name-or-number” with the cast member’s name or number in the cast window.*

---

## *Improving movie performance*

Movies may play back faster or slower than expected, depending on the complexity of the animation. When a movie has many sprites that move at once, large cast members, or stretched sprites, it will animate more slowly.

You can also expect speed differences when your movies are being played back on minimally configured PCs. A complex movie that plays at 30 frames per second on your 68040 Macintosh with 8 MB of RAM will not play back at the same speed on an 80386 16MHz PC with 2 MB of RAM. Some video cards are faster or slower than others, which also affects movie speed.

In general, Director runs faster when smaller sprites are used for animation, when fewer sprites are moving at the same time, or when you have fewer layers of sprites on the stage.

Here are some specific ways you can improve a movie's speed:

- ◆ For color animation, use 1-bit sprites wherever possible. One-bit sprites animate as fast as or faster than 8-bit sprites and they occupy less memory. Use the Transform Bitmap command on the Cast menu to reduce the color depth to 1-bit. Once the cast member is reduced to 1-bit you can still apply color. Just select the sprite on the stage and choose a color from the foreground or background color chips in the tools window.
- ◆ If you stretch or compress a sprite on the stage with Director's In-Between command (or manually by dragging a selected sprite's handle), you may find that the movement is slow and jerky, particularly if the sprites is large. To make the animation faster and smoother, use the Auto Distort command on the Effects menu to create a series of sprites that stretch or squeeze. Use the Cast to Time command on the Cast menu to place the sequence in the score.
- ◆ Give stationary sprites the copy ink, set trails on, and set the background color to the same color as the stage background. Your movie will run faster and use less memory.

- ◆ If you have a number of sprites on the stage and only one or two are moving, use a screen capture utility to create a stationary background that includes all the stationary sprites on the stage. Animated sprites move much faster across one large background than across a background composed of many individual sprites.
- ◆ Because QuickDraw sprites and text animate more slowly than their bitmapped equivalents, use the Convert to Bitmap command on the Cast menu to change QuickDraw sprites and text to bitmaps. Remember though that bitmaps use more memory than QuickDraw objects. Use your judgment about whether conserving memory or animation speed is more important.

---

## *Synchronizing graphics to sounds and digital video*

This section describes the two main strategies for synchronizing graphics with sounds or digital video: the tempo channel (for simple applications) and Lingo (for advanced applications).

Many variables influence the playback rate of Director movies. While synchronization of multiple graphic elements can be easily achieved in the score window, synching graphics to sounds or digital video can be difficult. Some factors are hardware-related: CPU speed, available RAM, and the playback medium (disk vs. CD-ROM). Other factors depend on your movie: color depth, stage size, sprite size and movement, transitions, inks, and the length and resolution of sounds or digital video. The simpler your movies are, the less work Director must do to draw and move complex graphics. Simpler movies allow Director to concentrate on playing sounds or digital video at the proper time.

Here are some general tips:

- ◆ Don't mix tempo channel events with Lingo. It's best to use just one of these approaches at a time. You may get unpredictable results if you combine tempo channel "wait" commands with Lingo timing commands in the same frame.
- ◆ Keep sound files small. This usually works best for voiceovers, sound effects, and music loops. For long musical selections, see "Sound-only QuickTime," later in this section.
- ◆ Lock the playback rate. Synchronization becomes a problem when moving movies to different machines. This is because sounds and digital video always play at the same speed, but graphics slow down or speed up depending on the speed of the CPU. You should always test your movie on the slowest possible machine, optimize for that platform, and lock your playback rate as a final step.

- ◆ Keep your System file clean. Timing of events can change even on the same machine, depending on how memory is allocated at the time. To get the most consistent playback, try to eliminate or turn off anything that might be interrupting or slowing down your system, including AppleTalk, file sharing, mail programs, large items in the Clipboard, programs or control panels open in the background, and system extensions.

---

## *Synchronizing with the tempo channel*

For the beginning user, the Set Tempo dialog box's Wait for Sound and Wait for QuickTime selections are the easiest way to pause a movie while sounds or digital video finish playing. But beware! These "wait" states are similar to the "pause" command that interrupts animations and scripts. Once you've entered the pause, animation stops and you lose interactive control.

You can also use the tempo channel to simulate playback on slower machines. For example, if your base platform is a slower Macintosh or a 20 megahertz 386, you may want to set your tempo to 4 to 6 frames per second, even though you might be able to play at 10 frames per second on a 68040 computer. That way you'll get better timing (but not necessarily the best possible performance) on faster machines. To get better performance on faster machines, see "Optimizing for individual machines," later in this section.

---

## *Synchronizing with Lingo*

Lingo can be used to:

- ◆ Measure time (`startTimer`, `timer`, `the time`, `the ticks`)
- ◆ Pause (`delay`)
- ◆ Change tempo (`puppetTempo`)
- ◆ Play sounds (`puppetsound`, `sound playFile`)
- ◆ Stop sounds (`puppetsound`, `sound stop`)
- ◆ Wait for sound (`soundBusy`)
- ◆ Play a digital video (`movieRate = 1`)
- ◆ Stop a digital video (`movieRate = 0`)
- ◆ Get the length of digital video (`duration of cast`)
- ◆ Get or set the time of digital video (`movieTime of sprite`)
- ◆ Set an in and out point for a digital video movie (`startTime`, `stopTime`).

A good way to maintain interactivity while waiting for sound to finish is:

```
if soundbusy (n) then go to the frame
```

Or, for digital video:

```
if the movieTime of sprite n < the duration of cast-  
  (the castNum of sprite n) then go to the frame
```

or:

```
if the movieRate of sprite n then go to the frame
```

For maximum performance, use the following alternate technique:

```
--in a previous frame
on exitFrame
--instead of calling the function over and over
--(a performance hit), you retrieve the value once
--and put it in a global variable for easy access
global gMoovLength
set gMoovLength=the duration of cast n
(the castNum of sprite n)
end
--then, in the frame you want to wait for the moov
on exitFrame
global gMoovLength
if the movieTime of sprite n < gMoovLength then
go to the frame
end if
end
```

This approach has the added benefit of allowing user actions such as mouse clicks or key presses to affect sprites on stage. (If you use the Wait for QuickTime Movie setting in the Set Tempo dialog box, Director stores any user actions until the movie finishes. This can produce unintended results.)

For even better digital video performance, use a repeat loop:

```
repeat while the movieRate of sprite n
    if the mouseDown then exit
    updateStage
end repeat
```

Lingo's repeat loops are optimized for fastest processing. This bypasses `exitFrame` and `enterFrame` calculations, on `idle` handlers, and system events such as a `keyDown`. The resulting playback will be as fast as in a non-interactive digital video playback application. Playing a digital video movie within a repeat loop overrides any interactivity not explicitly embedded within the loop (note the `mouseDown` command embedded in the above loop). Using the `movieRate` command assumes that the digital video movie does not initially appear on the stage in a paused state.

### *Preloading the cast*

You can use memory management techniques to enhance synchronization. Pauses often occur in playback when large graphics and sounds are being loaded into memory. Unloading unnecessary cast members and preloading sounds will minimize these interruptions. Take advantage of natural pause points in your movie to perform loading and unloading of cast members.

To see how memory is being used, place a text cast member on stage, name it "memoryCount," and put the following in a movie script:

```
on idle
    set the text of cast "memoryCount" to string -
        (the freebytes)
end idle
```

### *Optimizing for individual machines*

On a Macintosh, you can use Lingo's `machineType` function to determine what hardware is available for your movie. Depending on the speed of the machine, you can use `puppetTempo` to set the optimal playback rate for animation, and use `puppetSound` and `sound playFile` commands to launch sounds at the appropriate time.

Keep in mind that sounds and digital video will always play at the same speed on every machine (although QuickTime can display more frames in the same amount of time on a faster machine, the movie will be the same overall length).

### *Sound-only QuickTime movies*

Once a standard sound file (SoundEdit, AIFF, or resource) begins playback, Director has no way to track timing within the file. Using sound-only QuickTime movies is a good way around this.

By converting sound files to sound-only QuickTime movies, you can use Lingo to synchronize graphic events to sound. For example:

```
--after 6 seconds, make a sprite appear on stage
if the movieTime of sprite n > 360 then
    set the visibility of sprite 1 to TRUE
end if
```

Use > or < instead of = when checking the timing of QuickTime movies. It's unlikely that the proper line in your script will execute at the exact tick returned by the movie.

Be careful when creating sound-only movies. Often these movies are only pointers which reference the original file. Don't delete the original, and be sure to copy it when moving your movie to another volume. Better yet, make the movie self-contained (using the QuickTime Movie Converter or an application such as Adobe Premiere or VideoShop).

You can make a sound-only QuickTime movie that loops by setting the `startTime` and `stopTime` cast properties. You can set the `loopStart` and `loopEnd` points dynamically, and the escape from the loop can be triggered using Lingo (by a button, timer, or counter) rather than in the score.

For example:

```
--make sure that the sound-only digital video cast member is set to "loop" in the Cast Member Info dialog box
```

```
set the stopTime of sprite n to loopStop
```

```
if the movietime of sprite n > loopStart then set the startTime of sprite n = loopStart
```

```
if [some condition] then set the stopTime of sprite n = the duration of cast "mooV"
```

### *Maintaining synchronization in a long animation*

You can use Lingo to tell Director to drop frames in a long animation in order to maintain synchronization with a sound file or the internal clock.

In the frame where the animation sequence starts, use the following frame script:

```
global gStart, tps
--use your choice of frames-per-second:
set fps=10
--convert to ticks-per-second
set tps=60/fps
start Timer
--set the starting frame:
set gStart=(the frame)
--go to appropriate frame:
go to frame ((the timer)/tps) + gStart
```

Tweneed in the frame script of each frame in the sequence:

```
global gStart, tps
got to frame ((the timer)/tps) + gStart
```

Director will now delay or jump ahead in the animation to maintain synchronization.

To optimize synchronization, you can replace the variables with literal values. For example, suppose you want the animation starting at frame 200 to play at a constant 15 frames-per-second, regardless of the computer's speed or cast loading:

```
go to frame ((the timer)/4) + 200
```

---

## *Strategies for debugging Lingo code*

There are two good methods for debugging Lingo code:

- ◆ Use the message window
- ◆ Add script commands that display the status of variables and script execution.

An important part of debugging is being able to search through the code and keep track of what's supposed to happen in certain sections. You can use comments to let others know (or remind yourself!) how certain scripts or variables are being used, and you can use the Find/Change command on the Text menu to search for and replace text within a script, or in all scripts.

Here are some helpful debugging tips:

- ◆ Build and test your scripts incrementally. There's no need to create the entire script before you begin testing it.
- ◆ Use alert boxes as temporary substitutes for complex parts of a script. Clauses in an "if/then/else" construction can trigger an alert while you test the overall correctness of the "if" statement. You can later replace the alert boxes with the proper Lingo commands and test them separately.
- ◆ If a script doesn't work as you expect, try to isolate the elements that might be causing the problem. Remove new elements added since the last successful test of the script, and then gradually add them to the script until you encounter the problem. Or, make a smaller test movie that just contains the elements you think might be causing the problem.

## *Using the message window*

The message window is a real-time Lingo interpreter. You can type commands into the message window and see them execute in real time. For example, open the message window and type:

```
beep
```

Then press Return. This executes Lingo's beep command, triggering your computer's alert sound. You must press Return after every line for a statement to execute in the message window. The message window is a good place to experiment with scripting syntax. You can copy scripts that work in the message window and paste them into any script window. Conversely, you can copy scripts that don't work from any script window and test them in the message window.

All variables used in the message window are global variables. To determine the values of local variables within a handler, use the put command.

You can also use the message window to execute any custom handler defined in a movie script.

## *Using the put command*

Lingo's put command is one of the most useful commands to use in the message window. For example, by typing:

```
set x=1+1
```

```
put x
```

the message window will display the result (in this case, 2). The put command can be a bit confusing, since it can also perform operations similar to the set command: both can be used to put a value into variable. For example, these two statements:

```
set x=1+1
```

```
put 1+1 into x
```

are equivalent.

But the `put` command can also be used to test for the current value of a variable. At any point, you can use the `put` command in the message window and check variables. This can be especially helpful when working with global variables or arguments which are being passed between scripts.

You can also use the `put` command from within any script to send a message or a value to the message window. For example:

```
on checkDate
    set dateToday = the long date
    -- the next line puts the variable into the Message
    -- window
    put dateToday
    if dateToday contains "Saturday" then alert-
        "Working weekends again, are you?"
end checkDate
```

If it were Saturday and this script had not resulted in an alert, you could check the message window to verify the value of the variable `dateToday`.

When you have many variables to check and you're using multiple `put` commands, use the `&` text operator to add a string to each statement to help identify the values. For example:

```
on messageWindowDisplay
    set a = 1
    set b = 2
    set c = 3
    set total = a+b+c
    put "a=" & a
    put "b=" & b
    put "c=" & c
    put "total=" & total
end messageWindowDisplay
```

The message window displays the following:

```
--"a=1 "  
--"b=2 "  
--"c=3 "  
--"total=6 "
```

### *Displaying values on the stage*

You can also create your own message window on Director's stage by creating a text field to display the status of variables. For example:

```
put the mouseH into field "showMessage"
```

Eventually, however, this will affect performance.

### *Script comments*

It's a good idea to add comments to your scripts as you create and edit them. Any line that starts with a double-dash (--) is treated as a comment, and none of its contents are interpreted by Lingo. For example:

```
--Lingo doesn't care what I type here...  
--This tests whether the user has clicked the  
correct sprite  
--Sprite 2 is correct; sprite 3 is incorrect  
on testUser  
    if the clickon = 2 then go to frame "correct"  
    if the clickon = 3 then go to frame "incorrect"  
end testUser
```

If you ever have to revise a script that's over a week old or that was written by someone else, you'll be glad you left a reminder of what you were thinking about at the time you wrote the script.

## *Using Find/Change*

The Find/Change command on the Text menu can save time when you need to find and/or change the name of a variable, handler or filename across several scripts.

## *Common error messages*

When Lingo encounters an error, Director displays a dialog box that describes the error and also includes the section of code that generated the error. Note the position of the question mark in the code; it indicates the location of the error. Clicking the script button will take you directly to the error in the script so you can correct it. Here are some common messages:

- ◆ "Syntax error"  
This is often an error in spelling or punctuation. It can, however, indicate missing words or misused commands.
- ◆ "Handler not defined"  
This is usually caused by a previously unencountered word, or a “non-word” which Lingo is trying to interpret as a variable or handler. For example, you may have typed `puppet sound 0` (two words) instead of `puppetSound 0` (one word).
- ◆ "String value expected"  
This can happen when you pass a numerical value to a Lingo command or function that expects a string. Use the `string` function to convert the value to a string.
- ◆ "File not found" and other path errors  
Pathnames are case-insensitive and space-sensitive; being off by one letter will result in an error. It's best to use the Lingo `pathname` command when attempting to access external movies or sound files. This will automatically return the pathname to the current movie's folder. Another strategy is to copy the drive and folder names directly from your desktop, and paste them into the script. Or, in the message window, enter the Lingo command `put the fileName of cast n`, where *n* is the cast number of the linked file, and copy the pathname as needed.

A good way to catch errors before they occur is to watch your script formatting as you enter the script in the script window. Properly written scripts automatically format the indentations as they are entered. Improperly written scripts display flush left.

---

## Using 32-bit images

In addition to supporting 1-bit, 2-bit, 4-bit, and 8-bit color displays, Director supports 16-bit, 24-bit, and 32-bit color environments.

This section uses 24-bit and 32-bit interchangeably to mean the full range of 16.7 million colors that a Macintosh can display.

32-bit color is especially useful for displaying color photographs on screen. Note, however, that it is difficult, if not impossible to satisfactorily animate 32-bit images on any current Macintosh. See “Animating 32-bit images” later in this section.

For the most part, use Director with 32-bit color just as you would use it with 8-bit color, with a few exceptions.

► **Tip** *Since 32-bit images require a lot of memory, you may want to increase Director’s application memory size using the Get Info command on the File menu in the Finder.*

### *Hardware and system considerations*

Director supports configurations of one or more monitors. If you use more than one monitor and want to use 32-bit color, the stage must reside on a monitor controlled by a 32-bit video card in order to display 32-bit images.

**Note** *A 640 x 480 stage set to 16-bits requires approximately 600K of memory; 32-bits requires approximately 1200K of memory. Use the Monitors control panel to set the color depth of your monitor to display millions of colors.*

### *Importing 32-bit images*

Import a 32-bit image as you would any other image. Choose Import from the File menu to import a 32-bit PICT or copy a 32-bit PICT from the Clipboard and paste it into the cast.

Once the image is stored in the cast, it can be placed on the stage and edited in the paint window.

32-bit images require a lot of disk space and RAM. One way to import several 32-bit images at once is to link them to your movie using the Import command on the File menu, and click Import All in the dialog box. Make sure the Link to File checkbox is selected.

If you are editing 32-bit images in the paint window, use virtual memory if memory is low. If you are auto-scrolling a 32-bit image in the paint window and are low on memory, you can use the scrollbars instead to scroll your image.

## *Gradients*

When you're working with 8-bit or 4-bit images, gradients are always based on the current palette. This is not always true with 32-bit images.

If you want the full spectrum of colors (16.7 million) to be used when a gradient is created, use the “pattern” gradient methods. Pattern Best Colors and Pattern Best Colors See Thru create the smoothest possible gradient between the foreground and destination colors. These methods make use of as many of the available 16.7 million colors to create a smooth blend between the two colors.

Pattern Adjacent Colors and Pattern Adjacent Colors See Thru use as many colors as are necessary to create a consecutive spectrum of colors between the foreground and destination colors. The difference from the Pattern Best methods is that all color groups between the foreground and destination colors are used to create the gradient. Since the 32-bit environment colors are organized in the same order as the spectrum, the Pattern Adjacent methods are very useful for creating rainbow-like effects. To access the full range of the spectrum, use a bright red and a dark blue for the foreground and destination colors; the spectrum's colors are ordered as follows: red, orange, yellow, green, blue, indigo, violet.

Dither Best Colors, Standard Dither, and Multi Dither are the functional equivalents of Pattern Best Colors in a 32-bit environment.

All other gradient methods function the same as they would in 8-bit or 4-bit environments.

## *Using palettes and colors*

Palettes represent the set of colors that can be used with 4-bit and 8-bit images. This is because 4-bit (16 colors) and 8-bit (256 colors) video cards are only capable of displaying a limited number of colors at one time.

You cannot modify pure white or black (the first and last color positions in the System palette). You also cannot modify any of the built-in palettes that come with Director including the System palette. You can only modify duplicated or imported palettes.

32-bit images, however, do not use palettes. Instead, they get their colors directly from a video card capable of displaying 32-bit images. 32-bit video cards are capable of displaying up to the 16.7 million colors that a Macintosh can generate. To display all 16.7 million colors simultaneously, you would need a monitor that has 16.7 million pixels—4096 x 4096—and over 67 Megabytes of memory!

Therefore, 32-bit images are not associated with a palette, as an 8-bit image would be. In fact, 32-bit images ignore palettes set in the score, paint, or cast windows—they have no effect on the way the 32-bit image displays. The only relationship between 32-bit images and palettes is this: palettes are used as a means of selecting colors to edit and manipulate 32-bit images in the paint window. You can use as many palettes as you want to choose colors from; the display of 32-bit images is never affected by the current palette.

To quickly change palettes to choose from different sets of colors:

1. Open the paint and color palettes windows so you can see both windows simultaneously.  
The Paint, Effects, and Text menus appear.
2. Select the paint window.
3. To use the colors stored in another palette, choose a palette from the palette pop-up menu in the color palettes window.  
Choose the colors in the new palette from the color chips in the paint window. This procedure only works when you are editing 32-bit images.

### *Accessing colors not found in existing palettes*

You cannot modify any of the built-in palettes included with Director. To use a color other than the set of colors included in these palettes, you need to duplicate one of the palettes.

If you are modifying an existing built-in palette in the color palettes window, the palette is automatically duplicated when you change a color.

### *Animating 32-bit images*

Because of the high amount of processing power required to display 32-bit images, it is not always possible to animate 32-bit images and then play them back in real time. The best way to satisfactorily animate 32-bit images is to record them frame-per-frame to videotape.

To use 32-bit images that will be played back in real time, you can combine 8-bit and 32-bit images in one document. You may wish to use 32-bit images as static background images. For best performance, set these sprites to use the trails ink. You can then animate other 8-bit cast members in the same document.

To create and edit 8-bit images in a 32-bit document, switch the monitor's resolution to 256 colors. To edit 32-bit images again, you must set the monitor to millions of colors.

**Note** *Color cycling, palette transitions (including fade to black, and fade to white), and bit and pixel dissolves do not work in 32-bit environments. This is because these features require either that palettes to be in use or that a video card set to 256 colors be present.*

---

## *Creating movies for Windows*

For a movie created in Director, differences between Macintosh and Windows affect how the movie appears on Director for Windows. This section provides tips and guidelines for adapting your work in Director to create effective movies for Windows.

These guidelines don't replace the *Using Director* and *Using Lingo* manuals. Those two manuals provide many sample scripts and step-by-step authoring instructions.

### *Ensuring high-quality Windows movies*

Director for Windows brings the power of a Director movie to the PC. Of course, the Macintosh and the PC are different, so some aspects of a Director movie may perform differently on the two platforms.

You can ensure high-quality Windows movies by being aware of platform differences when you create your Director movie on the Macintosh. The following table lists the important platform differences and some solutions for working around them.

Difference	Solution
Memory differences between machines can affect how smoothly movies play back.	Optimize memory use. If possible, avoid large cast members such as large sounds and bitmaps; use linked cast members when possible. Control cast member loading to avoid noticeable pauses.
Color may differ between the two platforms. Windows sometimes translates Macintosh colors to colors that are similar but not identical.	Use palettes that give reliable color on the PC, such as the built-in System-Win palette. Use dithering and image conversion tools to translate colors. As you are authoring, frequently check the movie's colors on the PC.
Director for Windows does not mix sound. Only one channel can play at a time, and sound channel 1 takes precedence over sound channel 2.	Mix your two sounds using software like SoundEdit Pro.
Macintosh and PC monitors may be different sizes.	On the Macintosh, set the stage size small enough to fit on the Windows monitor.
Playback speed may vary on different platforms.	<p>When animation is too slow on the PC, try to use simple animation and fewer cast members. Avoid large QuickDraw and stretched sprites. Export frames with inks and then import the composited images.</p> <p>When animation is too fast on the PC, possibly the Macintosh you authored on is slower than the PC. Try lowering the tempo setting of the original Macintosh movie.</p>
QuickTime movies must be in the proper format to play back on the PC.	Convert QuickTime movies using the QuickTime Movie Converter. Director for Windows only supports direct to stage playback for QuickTime and AVI movies.

Difference	Solution
<p>Macintosh and DOS filename conventions are different. This may create problems in cross-referencing among movies after the movies are transferred to the PC.</p>	<p>Give Macintosh movies, linked cast members, and the folders they are in names that have eight characters or fewer, and use only characters that are valid for DOS. Do not use periods or blank spaces in the Macintosh name.</p>
<p>Macintosh and Windows fonts may differ.</p>	<p>Substitute available Windows fonts for Macintosh fonts, use bitmapped text, or create a font map table using the Movie Info command in the File menu.</p>
<p>PICT images are only supported by Director for Windows on PCs that have QuickTime for Windows installed.</p>	<p>When QuickTime for Windows is not installed on the PC, convert PICT images to DIB files using a utility like DeBabelizer.</p>
<p>Keys used to start and stop movies are different on the PC than on the Macintosh.</p>	<p>Use Windows key combinations to start and stop movies.</p>
<p>Under Windows, transitions may not occur at the same speed as on the Macintosh. Pixel dissolves are coarser under Windows.</p>	<p>Adjust transition speed when working on the Macintosh to compensate for the speed difference between platforms. Substitute other transitions for pixel dissolves if necessary.</p>

---

## Mapping Macintosh fonts to Windows fonts

This section describes how to map Macintosh fonts to Windows fonts if you plan to play back your movies on a computer running Windows.

Director stores font, size, and style information for each text cast member. However, when you open a movie created on a Macintosh using the Windows version of Director, Windows may create visually different text than the text displayed on a Macintosh.

Most Windows PCs don't use the same fonts used in your Macintosh movie. If Windows doesn't have one of these fonts, Director substitutes another available font.

Macintosh and Windows font sizes can differ. For example, 12-point text on the Macintosh can appear smaller on the PC.

When you create a new movie, Director looks for a file called FONTMAP.TXT in the same folder as the Director application. This file specifies how Director maps fonts between the Macintosh and Windows platforms. If Director finds this file, it uses it to create an internal font map for the movie. If no FONTMAP.TXT file exists, the new movie uses no font map. When you open the movie on the PC, Director uses the movie's internal font map to determine the appropriate substitute Windows fonts for text cast members that were created on a Macintosh. If the movie has no font map, Director substitutes other available fonts. The sample FONTMAP.TXT file at the end of this section provides an example of how Director maps Macintosh fonts to Windows fonts.

► **Tip** *If you want to guarantee that text looks identical on both platforms, convert your text cast members to bitmaps before opening the movie using Director for Windows. Be aware, though, that bitmapped text uses more disk space and cannot be edited as text.*

*Buttons, which appear as text cast members in Director's Text window, should not be converted to bitmaps. Bitmapping buttons removes certain properties or attributes assigned to button cast members and could affect Lingo scripts that refer to these button properties.*

## *Editing a movie's FONTMAP.TXT file*

When you open a *new* movie, Director looks for a font map file named FONTMAP.TXT in the same folder as the Director application. This file specifies the font mapping for all new movies. If Director can't find this file, it does not use a font map for new movies.

You can edit an existing movie's font map file to specify which fonts Director substitutes when you open the movie on the PC. Rather than creating this file from scratch, you can save the movie's internal font map table in a text file, and then edit this file as necessary.

To define the font mapping information for a movie, it's more convenient to edit the FONTMAP.TXT file before you begin authoring a movie, since Director automatically uses the information stored in the FONTMAP.TXT file when you open a new movie. (If you've already created the movie, you can still edit the font map file, but you will then have to manually load the file into the movie to have Director apply it to the movie.)

To define the font mapping information for a new movie:

1. Using any application that can edit text, open the sample FONTMAP.TXT file that's in the same folder as the Director application.

When you installed Director, this file was placed in the same folder as the Director application. If the file is missing, you can either re-install it or create it from scratch. See the end of this section for an example of a FONTMAP.TXT file.

2. For each Macintosh font remapping entry, type on one line:

```
Mac:MacFontName=>Win:WinFontName ↵  
[MAP (NONE|ALL)] [MACfontsize=>WINfontsize]
```

where *MacFontName* is the name of the Macintosh font, and *WinFontName* is the name of the Windows font being substituted for the Macintosh font.

The two arguments enclosed in brackets are optional. `MAP ALL` or `MAP NONE` specifies whether you want to remap characters with ASCII values greater than 127 or just pass them through. The default is `MAP ALL`.

You can specify how you want the characters to be remapped, as described in Step 3. The sample FONTMAP.TXT file contains mappings for a few commonly used graphical characters.

The last argument, `[MACfontsize=>WINfontsize]`, consists of one or more pairs of numbers, separated by a space, that let you map a Macintosh font size to a Windows font size.

Because font sizes appear smaller on a PC, you might want to map Macintosh font sizes to larger Windows font sizes.

3. For each Macintosh special character that you want to remap, type:  
Mac:=>Win: *OLDCHAR*=>*NEWCHAR* *OLDCHAR*=>*NEWCHAR* . . .

where *OLDCHAR* is the ASCII value of the Macintosh special character, and *NEWCHAR* is the ASCII value of the Windows character being substituted for it. You can enter as many remapping pairs as you want by separating each one with a space.

You can only remap characters whose ASCII values are greater than 127 and less than 255.

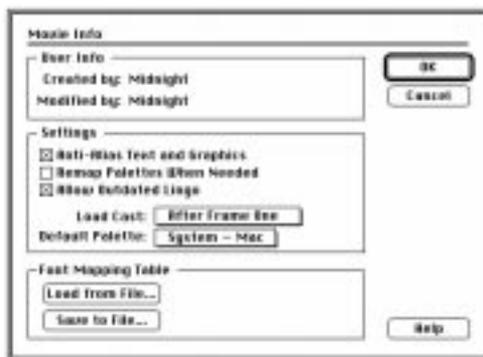
If you didn't specify `MAP ALL` for any of the font remapping entries, as described in Step 2, you can skip this step.

4. Save the file as ASCII text, in the same folder as the Director application.
5. Open a new movie in Director.  
When you open a new movie, Director looks for the font map file named `FONTMAP.TXT` in the same folder as the Director application. All new movies will use the font mapping information in the `FONTMAP.TXT` file. You can edit this file on a movie-by-movie basis, as necessary.

Existing movies continue to use the font map information (if any) stored within the movie rather than the font mapping specified in the FONTMAP.TXT file.

To change the font mapping for an existing movie:

1. Using any text editing application, edit the FONTMAP.TXT file as described in "Editing a movie's FONTMAP.TXT file," earlier in this section.  
Save this file using any name of your choice.
2. Open the movie whose font mapping you want to change.
3. Choose Movie Info from the File menu.  
Director displays the following dialog box.



4. Click Load from File.  
This option lets Director load the font mapping assignments specified in the font map file.
5. In the dialog box, select the font map file you just edited and click Open.
6. Click OK in the Movie Info dialog box.
7. Save the movie and close it.
8. Open the movie again.  
The movie now uses the font map information specified in the font mapping file.

Note

*If you edit a text cast member on the PC in a movie created on a Macintosh, the text cast member loses its original Macintosh font information. Similarly, if you edit the text cast member on the Macintosh for a movie created on the PC, the text cast member loses its original Windows font information. If you plan to edit a movie on both the Macintosh and Windows platforms, make sure that the font mapping file specifies that each Macintosh font has only one substitute font on the PC, and vice versa. This one-to-one font mapping ensures that Director will be able to assign the appropriate substitute font when you edit a text cast member on one platform and then open the movie on the other platform.*

A movie's FONTMAP.TXT file might look like this:

```
; This is a sample FONTMAP.TXT file
; Comments are denoted by using ";" or "--" to start the line
; The format for Font Mapping is:
; Platform:FontName => Platform:FontName [MAP (NONE | ALL)]
[OLDSIZE => NEWSIZE]

-- The format for specific Character Mapping is
-- Platform: => Platform:  OLDCHAR => NEWCHAR ...

; Here are sample mappings for the standard Mac fonts:
Mac:Chicago      => Win:"MS Sans Serif"
Mac:Courier      => Win:"Courier New"
Mac:Geneva       => Win:System Map All
Mac:Helvetica   => Win:Arial
Mac:Monaco       => Win:Terminal
Mac:"New York"  => Win:"MS Serif" Map None
Mac:Symbol       => Win:Symbol
Mac:Times        => Win:"Times New Roman" 14=>12 18=>14 24=>18
30=>24

; Here are sample mappings for the stock Windows fonts
Win:Arial        => Mac:Helvetica Map All
Win:"Courier"    => Mac:Courier
Win:"Courier New" => Mac:Courier
Win:"MS Serif"   => Mac:"New York" Map None
Win:"MS Sans Serif" => Mac:Chicago
Win:Symbol       => Mac:Symbol
Win:System       => Mac:Geneva
Win:Terminal     => Mac:Monaco
Win:"Times New Roman" => Mac:"Times" 12=>14 14=>18 18=>24
24=>30

; Note: From Windows to Mac, Courier and Courier New map onto
Courier. When coming back to Windows only Courier New will be
used.

; Here is a sample character mapping for the bullet char
Mac: => Win: 165=>149
Win: => Win: 149=>165
```

Note that:

- ◆ Comment lines must begin with two dashes (--) or a semicolon (;)
- ◆ Only one font mapping definition can be specified on a line.
- ◆ Arguments must be separated by spaces or tabs.
- ◆ If a font name consists of more than one word, it must be enclosed in quotation marks.

### *Using fonts on high-resolution screens*

Windows Setup program offers a Large font option, which uses a larger font on higher-resolution screens. When using the Large font option, text in TrueType or System font on an 800x600 pixel or larger display can wrap differently than it does on the Macintosh.

To avoid problems with the Large font option when playing back moves on screens larger than 800x600 pixels, use Windows' Small font option whenever possible. When it is necessary to use the Large font option, first test the movie on a large screen to verify that text looks appropriate.

---

## Managing PC memory

The amount of memory available on a PC can affect how the movie plays. Although a PC with a small amount of memory can play a large movie, limited memory slows down animation and causes a movie to pause frequently to load cast members.

- **Tip** *A good way to see how a Windows movie plays in a “worst case” situation is to restrict the amount of memory available on your PC, and then preview the movie. Restrict the available memory by creating a RAM disk that ties up free memory, so that the remaining free memory equals a worst-case PC. See the documentation that came with your PC for an explanation of how to create a RAM disk.*

---

## Getting good color

The typical color Macintosh has at least 256 colors available. PCs with VGA graphics offer only 16 colors; PCs with Super VGA graphics offer 256 colors.

Differences in color resolution between platforms can produce different colors on the PC than on the Macintosh. Several authoring techniques guarantee the truest possible colors on the PC. Which techniques you should use depend on whether you play back on 16-color PCs, 256-color PCs, or both.

## *Measuring color resolution and color depth*

Color resolution and color depth are often described in number of bits or in number of colors.

Number of bits	Number of colors
1	2
4	16
8	256
16	32,768
24	16.7 million

For example, black-and white images have only two color choices: black or white. This requires one bit to define each possible choice; this color depth is referred to as 1-bit. Four bits can define 16 possible color choices; this color depth is referred to as 4-bit.

## *Playing back in 16 colors*

Sixteen-color PCs are common. When Director for Windows runs a 256-color movie on a 16-color PC, each of the 256 colors is replaced by the most similar color on the 16-color PC. The replacement colors can vary from the original colors. The result is that image colors can change significantly and gradients become less smooth.

If your movie will be playing on a 16-color PC, you can guarantee good color by using Director's VGA palette. Any other 16-color or 256-color palette can be used; however, the movie's colors may be noticeably different on the PC.

Director's VGA palette uses only 16 chips, which duplicate Windows' 16-color VGA palette. Using these palettes on the Macintosh lets you work in colors that are identical to the colors used by VGA on the PC. The VGA palette has a limited choice of colors, though.

To use the 16-color VGA palette:

1. Set the color resolution of your Macintosh to 16 colors.
2. Select the palette channel in the score.
3. Use the Movie Info command to choose the VGA palette.

► **Tip** *Dithering images to the 16-color VGA palette while authoring on the Macintosh can improve the image's color quality when it is converted to 16 colors in Director for Windows. Use one of the third-party applications that converts 256-color images to 16-color images, or use the Dither option in the Transform Bitmap dialog box.*

### *Playing back in 256 colors*

On a 256-color PC, the System—Win palette produces accurate color. If you know that your movie will be played on 256-color PCs only, you can reliably use this palette in your movie.

Director for Windows also fully supports palette effects, such as color cycling and fades, on 256-color PCs.

---

### *Using palette effects*

Director for Windows supports all palette effects on 256-color PCs. No palette effects are supported on 16-color PCs.

On the PC, white and black can't be faded. When you use a fade to black in your movie, white pixels drawn with color 0 in the palette won't fade. Similarly, when you use a fade to white in your movie, black pixels drawn with the last color in the palette (255 in an 8-bit palette or 15 in a 4-bit palette) won't fade. If you need to fade white to black or black to white, create a custom palette that contains white in a palette position other than 0, and black in a palette position other than 255 or 15. These custom black and white colors will fade properly.

For small numbers of colors, color cycling set to 30 cycles plays faster on the PC than on the Macintosh. To ensure that color cycling is identical on the PC and the Macintosh, set color cycling to 29 cycles or lower when you create the movie.

---

## *Using sounds*

Like its Macintosh counterpart, Director for Windows will support two types of sound: internal sounds stored in the cast and linked external sounds that are played directly from disk.

Playing sound on Director for Windows requires that sound files are in the correct format and that the PC is properly configured for playing sound.

### *Using external sounds*

Director for Windows can play external sounds that are linked cast members or are controlled by Lingo commands. As long as these sounds files are available on the PC and the PC is configured for playing sound, Director for Windows plays them directly from the disk.

The external sounds must be in WAVE or AIFF format.

- **Tip** *Because linked sounds are played directly from disk, making large sounds into linked external sounds can conserve memory. For example, a 2 MB movie that uses a 1 MB sound has much more memory available when the sound file is played from a disk. The disadvantage to this technique is that the sound may drop out if you load another movie from the same disk while the sound is playing. To avoid this, put large external sounds on a separate disk. Keep in mind that external sounds are not included in a projector. You must supply them along with the projector.*

## *Using sound channels in the score window*

When playing stereo sounds that use both sound channels in the score window, sounds in sound channel 1 interrupt those in sound channel 2. For example, if you put a voice-over sound in sound channel 1, and then add a background sound in sound channel 2, the background sound will stop playing while the voice-over sound plays. In addition, when the sound in sound channel 1 finishes, Director restarts the sound in sound channel 2. However, linked stereo sounds or stereo sounds imported as a single cast member play back in stereo on Director for Windows.

- **Tip** *In Director, sound in the last frame of a movie continues to play or loop until the next movie begins or you exit the application. This sound can be a useful transition while Director loads the next movie. You can stop the sound using the Lingo `puppetSound 0` command, the `sound stop` Lingo command, or by using the MCI `stop` command when an MCI device controls the sound.*

## *Playing sound on a PC*

Unlike Macintosh computers, PCs do not always come with a sound card or the software to play anything other than a beep. The PC requires an additional sound card with the appropriate driver to play sounds. On a PC without sound capability, Director for Windows displays movie images but plays no sound.

If the PC contains a sound card but won't play sounds on Director for Windows, one of several things could be wrong:

- ◆ The sound card may have the wrong driver assigned. This could be true even though sound plays on other Windows applications such as MediaPlayer. To determine the right driver for your sound card, see the documentation that came with the card.
- ◆ An external sound file may be in the wrong format. Make sure that any linked external sounds for your movie are in WAVE or AIFF format after you transfer them to Windows. WAVE format is a PC-only format, while AIFF format can be used by Macintoshes and PCs.

For best results, use sounds sampled at standard sampling rates such as 11 kilohertz and 22 kilohertz.

## *Setting sound volume*

Sound volume settings on the PC work differently than on the Macintosh. On the Macintosh, a sound volume set too high produces a loud but clear sound. On the PC, high sound volume settings usually produce undesirable, distorted sound.

The sound volume in Macintosh movies is commonly set to the highest value in an attempt to get loud, clear sound. For best results when the movie plays back on the PC, avoid setting the sound volume with Lingo from within the movie. Instead, assume that the user has set the appropriate Windows sound volume. If you do set the sound level from within the movie, use a medium sound volume setting.

In Director for Windows, the sound volume setting of QuickTime for Windows movies is relative to the WAV driver sound volume setting. As a result, many QuickTime for Windows movies can play back louder or softer than the intended sound volume.

To control a QuickTime for Windows movie's sound volume when the movie plays back in Director for Windows, use the `volume of sprite` command.

---

## *Working with different size monitors*

A Director movie created on a Macintosh has the same pixel resolution when played on a Windows PC. Because the pixel resolution of PC and Macintosh monitors is sometimes different, movies can fill the screen or a window on the PC differently than they fill the Macintosh screen.

Using the Preferences command in the File menu, you can adapt the stage size for the PC monitors on which you expect to play back the movie.

- ◆ When the PC screen or window is larger than the stage, the area outside the stage displays the desktop.

- ◆ When the PC screen or window is smaller than the stage, the stage is cropped. If the movie has Always Center set in the Preferences dialog box, the stage is cropped around all four edges. If Always Center is not set, the stage is cropped on the right and bottom edges.

Remember that if the movie is played back in a window on the monitor, you can move and resize the window to accommodate the movie.

It is possible that the PC monitor or window will be smaller than 640x480 pixels. If you know that the movie will be played back on a smaller monitor or window, avoid cropping the movie by setting the stage size small enough to fit the PC monitor.

---

## *Using QuickTime*

Director for Windows supports QuickTime for Windows and Video for Windows movies effortlessly, as long as the files are in the correct format, use names that are valid for DOS, and are in a directory that is available to Director for Windows.

### *Converting QuickTime movies*

Apple's QuickTime Movie Converter supports all playback-related features and produces a Windows version of the QuickTime movie.

QuickTime movies used with Director for Windows must be converted using the Movie Converter in QuickTime.

To convert a QuickTime movie using the Movie Converter:

1. Open the movie in Apple's QuickTime Movie Converter.
2. Choose Save As from the File menu.
3. Enter the name of the movie.
4. Click the "Make movie self-contained" radio button.
5. Select the "Playable on non-Apple computers" checkbox at the bottom of the dialog box.
6. Click Save.

The converted QuickTime movie plays back on either Macintosh or Windows. You can still edit the original QuickTime movie on the Macintosh, but the changes won't appear in the Windows movie. To change the Windows version, make changes to the original QuickTime movie and convert the movie again.

You can also use Microsoft's utility for converting QuickTime movies into AVI movies, which can be played back using Director for Windows. If you use AVI, that the converted AVI files may be several times larger than the corresponding QuickTime for Windows files.

### *Importing QuickTime movies*

In Director, import the QuickTime movie into the cast. See the *Using Director* manual for an explanation of importing QuickTime movies. Since QuickTime movies are imported as linked files, you must transfer QuickTime movies separately to the PC in order to have them play in a Director movie.

#### Note

*When you change the QuickTime movie's filename to make it DOS-compatible, be sure to also update any references to the QuickTime movie within the Director movie.*

---

## *Using Microsoft Video for Windows (AVI)*

Director for Windows also supports Microsoft Video for Windows (AVI) movies, which end with the file extension *.AVI*. For Director for Windows to play an AVI movie, the AVI movie must be in the Director movie's DOS path.

If you are authoring a Macintosh movie for playback on a Windows PC using Video for Windows, give the QuickTime movies valid Video for Windows filenames before importing them into Director on the Macintosh, for example, "myMovie.AVI".

---

## *Naming files and folders*

Macintosh and DOS have different filename and folder name conventions. In DOS, the filename or directory name must be eight or fewer characters followed by an optional three-character extension (referred to as 8.3 format), and not use any of the following characters.

---

asterisk (*)	less than (<)
backslash (\)	period (.)*
brackets ([])	plus sign (+)
colon (:)	question mark (?)
comma (,)	semicolon (;)
double quotation mark (" )	slash (/)
equal sign (=)	space ( )
greater than(>)	vertical bar ( )
at sign (@)	characters with ASCII values less than 32

---

\* DOS interprets periods differently than the Macintosh does. In DOS, a period indicates the beginning of the file extension. This causes difficulty when names are converted for Windows.

When playing back movies that refer to other files—such as linked cast members—Director for Windows must be able to identify and locate movies and external resources. The DOS pathnames for these external files must follow the standard DOS filename conventions.

If you must change the name of a linked file to make it DOS-compatible, be sure to also update any references to the linked file within the Director movie.

When you transfer a Macintosh file to the PC, DOS automatically truncates the filename so that it conforms to the 8.3 format.

► **Tip** *The best way to ensure that Director for Windows locates movies and external resources is to assign Macintosh files and folders names that are acceptable to DOS. That way, the filenames and pathnames are consistent and usable when your movie plays back on the PC. Director for Windows does not convert filenames.*

For example, if you have a movie titled “Mayfair in Springtime,” you can use the filename MYFRNSPR.DIR. A movie that refers to your Mayfair in Springtime movie has a line of Lingo such as the following:

```
Play Movie "MYFRNSPR.DIR"
```

When playing back the movie, Director for Windows looks for the file MYFRNSPR.DIR when it encounters the `Play Movie` command. If you changed the name of the external resource to something other than what Director expects, the movie cannot locate the external resource.

Note *Director expects to find the shared cast movie under the name SHARED.DIR. Do not rename the shared cast movie and only use the SHARED.DIR filename for a shared cast movie.*

### *Choosing filenames and folder names*

Follow these rules when assigning names to Macintosh files and folders:

- ◆ Use eight or fewer characters, followed by an optional period and a three-character extension, for example MYMOVIE.DIR.
- ◆ Don’t use a period within the Macintosh filename or folder name.
- ◆ Don’t use any of the characters that are unacceptable in DOS filenames or directory names. Unacceptable characters are listed in the previous section.
- ◆ Double-check DOS-style filenames to make sure they aren’t already used before transferring files to the PC.
- ◆ Don’t start filenames with numbers.

For example, these Macintosh filenames can be used on the PC: MYMOVIE, MOVIE1, MOVIE2.

These Macintosh filenames would not work on the PC:

- ◆ FINALMOVIE, which contains more than eight characters
- ◆ My Movie, which contains a space
- ◆ MOVIE/1, which contains a slash, an unacceptable character
- ◆ INTRODUCTION.FINAL, which contains too many characters and an incorrect file extension.

### *Assigning pathnames*

Director uses pathnames to locate files for external resources—such as linked cast members—and in Lingo statements that refer to external files.

Pathname conventions on the Macintosh and the PC are different:

- ◆ DOS directory names must have eight or fewer characters and not use any of the characters that are unacceptable in DOS.
- ◆ DOS pathnames use slashes to separate directory names; Macintosh pathnames use colons to separate folder names.

For example, the Macintosh pathname “Hard disk:Director:Projects:Movie 1:Sounds” is equivalent to the DOS pathname “C:\DIRECTOR\PROJECTS\MOVIE1\SOUNDS”.

To be available to the movie on the PC, external files must be where Director for Windows expects to find them using the DOS pathnames you specified. Do the following to make sure pathnames are usable on the PC:

- ◆ Give Macintosh folders names that are acceptable in DOS. This guarantees that the directory always has the name Director for Windows looks for on the PC.
- ◆ Duplicate the folder structure between the movie and external resources on the PC. Directories do not have to be on drive C, but they should be on the same drive.

Also, the directory structure on the PC does not need to duplicate the entire absolute pathname that folders had on the Macintosh, but the structure must duplicate the relation between the files.

- ◆ Don't use periods in Macintosh folder names.

### *Using relative pathnames in Lingo*

If you move your files to a PC, the simplest way to organize them is to keep all linked files in the same directory as the movie file. This lets you avoid using relative pathnames in Lingo.

If you want to organize your linked files and movies in separate folders, you must use relative pathnames in Lingo to indicate the location of the files and folders.

Director uses the @ symbol, followed by a pathname separator character, to specify the folder in which the current movie resides. Use one of the following characters for the pathname separator:

- ◆ / (forward slash)
- ◆ \ (backward slash)
- ◆ : (colon).

For example, if you are in the root (top level) directory, and you want to access a file called Newmovie in the Final directory, refer to the file as “@\Final\Newmovie,” “@/Final/Newmovie,” or “@:Final:Newmovie.”

You can use the pathname separator character instead of a folder name, to indicate “backing up” a level in the directory structure. The following characters all indicate backing up one level:

- ◆ / or ../ (forward slash or double dot forward slash)
- ◆ \ or ..\ (backward slash or double dot backward slash)
- ◆ : (colon).

Some examples of relative pathnames:

---

Relative pathname	Description
MyFolder @:MyFolder	These directory names are equivalent. The MyFolder folder is a subdirectory of the current folder.
@:MyFolder:Newmovie @\Final\Newmovie	These file names are equivalent. The Newmovie file is located in the MyFolder directory.
@::Newmovie @\Newmovie	These file names are equivalent. The Newmovie file is located up one level from the current directory.
@:::Project:Intro:OtherFolder: Newmovie	The Newmovie file is located up two directories and down three directories from the current directory.

---

## *Inks and performance*

Director for Windows supports all inks that are available for Macintosh movies.

On both the Macintosh and the PC, Director draws some inks faster than others. Where speed is crucial in your movie, try to use the inks that are drawn fastest.

These inks are drawn at essentially the same speed as they are drawn on the Macintosh. You can rely on these inks in parts of a movie that require high-speed animation.

---

Copy	Ghost
Transparent	Matte
Background Transparent, when white is the transparent color	Reverse

---

These inks are fairly efficient, but require more time to draw. You can use them sparingly in animation without seriously affecting speed.

---

Not Copy	Background Transparent, when white is not the transparent color
Not Transparent	Mask
Not Reverse	Any ink effect applied to any 1-bit bitmapped sprite with colorization
Not Ghost	

---

These inks are slowest, and unless the sprite is very small or not moving, it is best to avoid them.

---

Add	Blend
Add Pin	Lightest
Subtract	Darkest
Subtract Pin	

---

---

## *Using Lingo commands*

Director for Windows supports all Lingo commands except those used to control Macintosh-specific features.

### *Unsupported Lingo commands*

Director for Windows doesn't support these Macintosh-specific Lingo commands:

- ◆ `closeDA`
- ◆ `openDA`
- ◆ `printFrom`
- ◆ `showResFile`
- ◆ `showXLib`

## *Partially supported Lingo elements*

Element	Comments
<code>Command-Period</code>	The PC's Escape key is similar to the Command-Period, Command-Q, and Command-W key combinations on the Macintosh. Pressing Esc on the PC halts a movie and returns the user to the Director for Windows application window. You can disable this Esc key function by setting the Lingo property <code>exitLock</code> to true.
<code>alert</code>	The <code>alert</code> command doesn't sound a system beep when displaying an alert box under Windows.
<code>colorDepth</code>	Color depth can be read on the PC, whereas on the Macintosh, it can be both read and written. If you attempt to give this property a value on the PC, an alert appears.
<code>commandDown</code>	The <code>commandDown</code> function always returns False on the PC.
<code>exitLock</code>	Setting the Lingo property <code>exitLock</code> to true prevents halting a movie when the Esc key is pressed; on the Macintosh, setting <code>exitLock</code> to true prevents halting the movie when Command-period, Command-W, or Command-Q is pressed.
<code>keyCode</code>	<p>When a key is pressed, Lingo can return two values: the character code and the key code. The character code is the alphanumeric value of the key that was pressed; the key code is the numeric value that identifies the specific key that was pressed. The <code>keyCode</code> function is useful for detecting when the user has pressed the arrow keys, function keys, or other nonalphanumeric keys.</p> <p>Key codes on the Macintosh and the PC are different. Also, key code values on PC keyboards vary from manufacturer to manufacturer. Check key codes, especially for function keys, on your delivery platform.</p> <p>The character code can be found by using the Lingo function <code>key</code>. The value returned from the <code>key</code> function is the last character typed—for example, the letter <code>k</code>.</p>

Element	Comments
<code>machineType</code>	The <code>machineType</code> function returns the value 256 for any PC, regardless of type, processor, or manufacturer.
<code>movie</code>	The <code>movie</code> function returns the current name of the movie.
<code>open</code>	The Lingo command <code>open</code> launches other Windows applications. If the application's directory is not listed in the DOS path, you must provide the full DOS pathname. The filename must include the application's three-letter extension.
<code>optionDown</code>	The <code>optionDown</code> function returns the state of the Alt (alternate) key.
<code>pathname</code>	The <code>pathname</code> function returns the current DOS directory pathname. The pathname is returned in the Macintosh format, with all slashes replaced by colons.
<code>perFrameHook</code>	The <code>perFrameHook</code> property is not supported for subframes.
<code>restart</code>	The <code>restart</code> command behaves like the <code>quit</code> command: the current movie stops playing, then Director for Windows closes and returns you to Windows.
<code>shutDown</code>	The <code>shutDown</code> command stops the current movie, closes Director for Windows, and exits Windows.

### *Using Lingo properties*

Director for Windows supports all Lingo sprite properties, including `locH`, `locV`, `stretch`, and `ink`.

Most system properties are supported, including all event polling scripts, such as `keyDownScript`, `mouseDownScript`, and `timeOutScript`. However, system properties that are closely tied to the Macintosh hardware are not supported.

---

## Using MCI

Media Control Interface (MCI) is an interface between the PC and external devices such as laserdisc players, compact disc audio players, and other computer-controllable external devices. MCI commands allow the movie to control these devices when it is played back on the PC.

### *Sending MCI commands*

Director for Windows uses the Lingo command `mci` to pass MCI commands to Windows' MCI command interpreter. Director ignores these commands when the movie is played back on the Macintosh, but Director for Windows uses them when the movie plays back on the PC.

Each MCI command has the following syntax:

```
mci expression-string
```

Where *expression-string* represents any expression that causes a command string to be sent to the Windows MCI command interpreter.

The command string is typically one of the MCI commands. For a complete list of MCI command keywords, see the documentation that comes with the *Microsoft Multimedia Development Kit*.

For example, this script uses MCI commands to play the sound file called `CLOCK.WAV`:

```
mci open CLOCK.WAV type waveaudio alias sound
mci play sound
mci wait sound
mci close sound
```

You can use Lingo's `the result` function to display the value that the command string returns.

For example, this script displays the name of the first MCI device:

```
mci sysinfo all name 1
put the result
```

This sample script displays the number of available MCI devices:

```
mci sysinfo all quantity
put the result
```

### *Examples of MCI commands*

The following examples show the use of MCI commands you can include in a Director movie:

- ◆ The `mci "play devicewait"` command pauses the playback head until the device name specified by *device* completes its task.

- ◆ The `mci Play Wave` command plays a WAVE file:

```
-- Put the following in a button script
mci Open c:\windows\tada.wav type waveaudio-
alias wave

mci Play Wave

-- Put the following wait in the script channel
mci Wait wave

-- Do this after wait
mci Close wave
```

- ◆ The `mci status` command checks to see whether a WAVE file is still playing:

```
-- Check for wave done
mci status wave mode
if the result = "stopped" then
-- wave file is stopped
end if
```

- ◆ The `mci close` command closes an MCI device when you are finished with it.

You must close an MCI device after it is finished playing. This is particularly true for a waveaudio device. If you don't close an MCI device after it is finished, an attempt to play a second WAVE file while the first one is still active will fail.

This `mci close` command closes a wave audio device:

```
mci close wave
```

This `mci close` command closes all currently open MCI devices whether opened by Lingo or not:

```
mci close all
```

---

## *Working with XObjects*

For cross-platform portability, you should store XObjects in separate files, not in the Macintosh resource fork.

At startup, Director for Windows looks for and executes a file named LINGO.INI. You can edit this file to include any Lingo commands you want to execute at startup. For example, you can include commands that instruct Lingo to open specific XObjects.

When working with custom hardware that can't be controlled by MCI commands, it may be necessary to write a special XObject to control the device properly. For information about creating XObjects, see the *XObject Developer's Kit*.

---

## *Using PICT images*

There are two ways to include a linked external PICT image in your Windows movie:

- ◆ If QuickTime for Windows is available, the PICT image can be played as it is.
- ◆ If QuickTime for Windows is unavailable, PICT images on the Macintosh must be converted to DIB format and transferred to the PC. Director for Windows can then use the DIB file. Use one of the several third-party image conversion tools to convert PICT files to DIB files.

▶ **Tip** *The PICT image quality is higher when displayed using QuickTime for Windows rather than as a DIB file. Consider using QuickTime for Windows if it is available on the PC.*

---

## *Linking movies*

Director for Windows supports all `play` and `go` command variations. Using the `play` and `go` Lingo commands, you can link several movies into one seamless presentation that plays back on Director for Windows.

This is a sample Lingo script that uses `go` and `play` commands to link movies:

```
go to movie "Desert.DIR"  
go to frame 23 of movie "Chicago.DIR"  
go to frame varFrame of movie varMovie  
go to frame "main loop" of movie "Intro.DIR"  
play movie "Desert.DIR"  
play frame 23 of movie "Chicago.DIR"  
play frame varFrame of movie varMovie  
play frame "main loop" of movie "Intro.DIR"
```

Note the use of DOS filenames in this script. For information about DOS file naming conventions, see “Naming files and folders,” earlier in this manual.

---

## Creating menus

Director for Windows supports menu bars, standard pull-down menus, and menu scripts. Use the `installMenu` Lingo command to add menu bars to your presentation.

However, Director for Windows doesn't support all menu features. The following table lists which special character options are and are not supported for the Lingo keyword menu.

Description	Supported
Command key equivalent	N
Menu scripts	Y
Apple menu symbol	Y*
Disabled menu item	Y
Disabled menu line	Y
Checkmarked menu item	Y
Bold style	N
Italic style	N
Underline style	N
Outline style	N
Shadow style	N

\* On Windows, the Macromedia logo is used in place of the Apple symbol.

---

## *Features that behave differently on the PC*

This section describes Director features that behave differently on the PC.

### *Using stretched sprites*

Just as on the Macintosh, stretched sprites under Director for Windows can slow down your movies, especially on slower PCs. However, under Windows, large stretched sprites slow down playback more dramatically than on the Macintosh.

If you must stretch a sprite, consider stretching the sprite in the paint window and replacing the one on stage. If you stretch a sprite on the stage to use as a movie backdrop on which other sprites are animated, playback will slow down dramatically. To avoid this, create backdrops in their final size.

### *Controlling external applications*

Director for Windows doesn't support Windows' object linking and embedding (OLE) or external application control through Windows' dynamic data exchange (DDE). However, you can open another Windows application using the Lingo command `open`. The `open` command looks up the application in the current path as set by the `PATH` environment variable.

---

## *Transferring files to the PC*

As you develop Director movies on the Macintosh, you will want to preview your work on the PC frequently. Previewing lets you know exactly how the movie plays on the PC as work progresses.

There are three basic ways to transfer files:

- ◆ Over a network
- ◆ Through a direct connection, such as MacLink Plus, between the Macintosh and the PC
- ◆ On removable disks and cartridges that you carry from one computer to the other.

The most practical way to transfer files depends on how large the files are.

Networks transfer files fastest. When you are working with large movies, consider putting the Macintosh and the PC on the same network. Using a network also lets you keep your Macintosh and converted Windows movies in the same folder structure on the same disk. This way, you can preview and edit movies in the same place without copying them from the Macintosh to the PC.

If your movie fits on one removable cartridge or disk, that method is usually adequate. Because the format for media differs between the Macintosh and PC, the Macintosh and the PC may require software that allows one computer to read media in the other computer's format.

**Note** *Many file transfer products have options for translating files into different formats. Do not use these translation options when transferring your movie.*

The following sections describe some common third-party products that are available to help you transfer converted movies from your Macintosh to a PC.

### *MacLink Plus/PC*

A complete kit for transferring and converting files between a Macintosh and a PC.

### *LapLink Mac II*

A complete kit for transferring files between a Macintosh and a PC or between an unlimited number of Macintoshes on an AppleTalk network.

### *DOS Mounter*

A software product that allows you to mount MS-DOS formatted cartridges or 3.5-inch disks on your Macintosh desktop.

### *TOPS*

A software product for the Macintosh and a software/hardware product for the PC that allows both platforms to be used as file servers and clients over LocalTalk and other networks.

### *Apple File Exchange*

A file translation utility that ships as part of Apple's system software.

### *PhoneNET PC*

A software product for high-volume data transfer between a Macintosh and a PC.

### *Timbuktu*

A software product that connects a Macintosh and a PC, transfers files between the two, and lets you control one from the other.

### *Apple PC Exchange*

A software product that allows you to mount MS-DOS formatted disks on your Macintosh desktop.

---

## *Frequently asked questions*

*What's the difference between the registration point of a cast member and the stage coordinates (locH and locV) of a sprite?  
How can I accurately position cast members on stage?*

For each cast member, Director uses a specific point, called the registration point, to determine the cast member's placement on the stage. Stage coordinates refer to the location where a sprite's registration point will be positioned on stage at a particular frame. These coordinates are the locH (measured horizontally from the left side of the stage) and the locV (measured vertically from the top).

- ◆ To see the registration point, open the paint window and click the registration tool. Use this tool to reposition the registration point. For accuracy, use this tool while zoomed in.
- ◆ To automatically center a bitmap's registration point, double-click the registration tool.
- ◆ QuickDraw cast members use the upper left corner of the bounding rectangle as their registration point.
- ◆ To precisely adjust a cast member's location on the stage, look at the stage coordinates in the score window, using the Extended view display mode. Select the sprite in the score window and use the keyboard arrow keys to move it a pixel at a time. You can also use the Sprite Info command in the Score menu to set the location.

*If I change the color depth of a cast member to black and white, colors lighter than medium grey change to white, and other colors change to black. How do I convert a color cast member to black and white, and how do I apply a dithered pattern?*

Often you must convert color cast members to conserve memory, or to adapt a color movie for playback on black and white monitors.

One simple way to get dithering is to use the Dither option in the Transform Bitmap dialog box. Alternatively, you can copy the cast members to a scrapbook, change the monitor's display mode to black and white, and then import the scrapbook back into the movie (or paste individual cast members into the cast window).

If you want more control over the subtleties of the dither pattern, use this technique:

1. Copy the cast member to the clipboard.
2. Open a grayscale graphic editor like Adobe PhotoShop.
3. Paste the image into a new window. This will convert the image from color to grayscale.
4. Adjust the image's brightness and contrast for best results.
5. Export the image as a 1-bit PICT (using the old format of PICT, before color PICT files).
6. Import the PICT image as a cast member in Director.

### *How can I do color cycling with 24-bit images (or in 24- or 32-bit monitor display mode?)*

Color cycling is an effect that depends on color position in a specific indexed palette. Images with color depths higher than 8-bits are not mapped to a palette, but instead use “direct color” RGB values representing each pixel of the bitmapped image.

If you are displaying 8-bit cast members in 24-bit or 32-bit modes, use the following technique, which emulates the color cycling effect by creating a series of duplicated palettes:

1. Temporarily switch the monitor to 8-bit display mode.
2. Select the same range of colors in each duplicate.
3. Use the Rotate Colors command on the Palette menu to shift colors one position to the left.
4. In the second duplicate, choose the Rotate Colors command twice.
5. In the third duplicate, choose the Rotate Colors command three times.  
Continue this procedure for each duplicate palette.
6. Duplicate the cast members that you want to use for color cycling, and re-map them to the duplicated palettes.
7. Create a film loop of the duplicated cast members.
8. Switch the monitor back to 24-/32-bit display mode.

### *How do I in-between colors in the score?*

Use the color chips in the tools window to tween a cast member’s foreground color. You can tween the foreground and background values between key frames. The tweening will cycle between the color positions you select within the active palette.

Blending colors in a custom palette will make the tweening affect appear smoother.

Use the following technique to tween 1-bit (black and white) cast members:

1. To assign the first color, open the score and select the cell of a cast member in one frame.
2. Open the tools window.
3. Click either the foreground or background color chip in the tools window, and choose the first color to be assigned from the palette.
4. In the score, select the next key frame, and assign it a color using either the foreground or background color chips in the tools window.
5. In the score, select the cells from the first key frame to the last key frame.
6. From the Score menu, choose In-Between Linear.

*The animation is running very slowly. How can I speed it up?*

If you have many objects which don't move on the stage, take a screenshot of them and paste it into the movie as a new cast member. This will improve playback speed, since Director will not have to process the overlaps and inks for each frame.

Here are some other things you can do to improve playback speed:

- ◆ Allocate more memory to the Director application
- ◆ Use bitmapped sprites instead of QuickDraw (shape) sprites
- ◆ Decrease the color depth of cast members to 8-bits
- ◆ Don't use stretched sprites. Use the Transform Bitmap command on the Cast menu to resize the cast member.
- ◆ For cast members that use inks, apply inks that animate more quickly, such as copy or matte.

*I am creating a kiosk in which I must interrupt the movie every 20 minutes and flash a message that asks the user to take a seat. How do I do this using Lingo?*

Use Lingo's date and time functions to control an interruption in your movie. You compare the current date and time with a table of values, and when they fall within a certain range, you can specify a change in the behavior of the movie.

Note *The time function returns a string. You must use the chunking functions to return an item within a text field, and then convert it to a number using the value function.*

Alternatively, you can send a timing signal at 20-minute intervals to one of the serial ports. A script in the kiosk would have to periodically branch and use the SerialPort XObject to check whether a character had been received on the port, and react accordingly.

*How do I use 4-bit cast members in the 8-bit monitor display mode? The colors change when I switch to 8-bit display mode.*

You must remap the 4-bit cast members to an appropriate 8-bit custom palette as follows:

1. If you have not already done so, install the Scrapbook FKey that comes with Director.
2. While in 4-bit display mode, use the Scrapbook FKey to take a screenshot of each cast member displayed in the correct colors.  
This creates a special Anim scrapbook in the System folder.
3. Switch the monitor's display mode to 8-bit.
4. Launch Director, and import the Anim scrapbook, importing the custom palette for each cast member.
5. Discard the Anim scrapbook from the System folder.

*In my movie, I swap cast members and use a linked AIFF sound every time the user clicks on a sprite. Why is performance so slow?*

You may not realize that you are asking the disk drive or CD-ROM to do two things nearly at once. Playing a short linked AIFF sound file while swapping cast members may cause the disk head to try to read data from two places in quick succession. This can cause performance to suffer.

The solution is to import the sound without linking to it, and use it as a puppet sound, which plays entirely in memory.

You can gain other performance improvements by using these standard memory management techniques for your cast members:

- ◆ If a cast member is a solid color, you can change the cast member's bit depth to black and white, and colorize the sprite using the tools window.
- ◆ If a cast member is rectangular, use copy ink instead of matte ink. Copy ink improves animation speed and uses less memory than matte ink.

*Imported PICS files are leaving trails on the stage. How do I correct this?*

When imported into Director, Mathematica and Swivel PICS files use the trails effect. PICS files are usually saved as a differential PICS files, so that only changes between frames are saved. Director automatically adds trails when you import the PICS file so that the bitmap image that makes up the PICS sequence appears correctly on the stage.

In addition, Mathematica PICS files include an unnecessary bounding rectangle which should be deleted.

To correct these problems:

1. Import the PICS file into Director.
2. Select the entire range of cast members in the score.
3. Choose no trails by removing the check from the Trails checkbox.
4. If this PICS was created by Mathematica, delete the unnecessary bounding rectangle cast member from both the cast and score windows.

*How can I control whether Director imports custom palettes when I import a PICT or PICS sequence?*

To have Director install the custom palettes in the score, select a cell in the palette channel in the score before importing the PICT or PICS sequence. If you don't first select a palette channel cell before importing the PICT or PICS sequence, Director doesn't include the custom palettes.

*When I use the Darkest or Lightest ink, sometimes I get unexpected results. How can I fix this?*

In very rare instances you may experience the results of bugs in the Macintosh's QuickDraw routines. These occur when you use the Darkest or Lightest inks. For example, mixing colors with Darkest or Lightest inks will not work well if one of the colors is a gray pattern. Another anomaly can occur when you use the air brush's speckled paint option to spray with the Darkest or Lightest inks. The size of the large drops of speckled paint will not be as large as they are with the other inks. If you encounter these problems, use a solid gray rather than a pattern to mix colors.

Inks and other palette effects often work more reliably when the monitor's color depth is set to 8-bits or less.

### *How can I play asynchronous sound in a digital video movie?*

To play a digital video movie's sound asynchronously, you must create two copies of the movie – one for the sound and one for the animation.

1. Select the digital video cast member in the cast window, and choose Duplicate Cast Member from the Cast menu.
2. Select the original digital video cast member, and choose Cast Member Info from the Cast menu.
3. Check Direct to Stage and Play Every Frame.
4. Choose the desired tempo setting and click OK.
5. Select the duplicate cast member, and choose Cast Member Info from the Cast menu.
6. Deselect the Video checkbox.  
Make sure the Sound checkbox is checked, so that only the sound portion of the movie plays.  
  
Check Loop if you want the sound to play in a continuous loop. Click OK.
7. In the score window, set each movie to appear in the same frames in separate channels.

### *How can I use different digital video tempos each time I loop a digital video movie?*

To use a different tempo each time you loop a digital video movie, use a fixed frame rate and then use the Lingo command the `movieRate` of `sprite channelNumber` to reset the rate each time you play the digital video movie.

### *How can I create palette transitions in a digital video movie?*

Since the Director movie is still running while the digital video movie is on the screen, you can put the palette transitions in the score.

*My movie uses a large digital video cast member. How can I edit the movie without having to always wait for Director to load the large digital video cast member?*

Create a small digital video movie and substitute it for the larger one. To substitute the smaller movie in the cast, open the digital video cast member's Cast Member Info dialog box, and click the linked file name at the bottom of the dialog box. Locate the small digital video movie, and link it into the cast. When you finish editing the movie, you can link the large digital video cast member back into the cast using the same procedure.

*How do I play a protected movie?*

The only way to play a protected movie is by calling it from a projector using the Lingo `go to` or `play` commands.

To create a protected movie, choose Update Movies from the File menu and click the Protect Movies checkbox.



My protected movie

Protected movies cannot be reopened in Director, so you should create them late in the project and keep a separate copy of your original (unprotected) movie. A protected movie no longer includes the Lingo source or cast thumbnails, so even if users try to open it with a resource editor, they will not be able to see how your movie was created.

If you protect a movie that has a .DIR extension, Director changes the extension to .DXR. Protected movies can be played on either platform; they must have the .DXR extension for Director for Windows to find them.

## *How do I use a projector?*

A projector is a play-only version of a Director movie that can be played by double-clicking it. Users do not need to have the Director 4.0 application to play a projector movie.



You cannot play a projector from another projector, nor can you play a projector from a Director movie. Projectors can only be played on the platform on which they were created.

The actual file size of a projector may be smaller than the total size of the movies in the play list, since Director compacts the movies when it creates the projector.

The default free memory for a projector is 4 Megabytes. You can create projectors that run in 2 or even 1 Megabyte of RAM. However, you should test them to make sure your movies run correctly. Projectors crash if they don't have enough memory allocated in which to run.

A projector's play list must include at least one movie. You cannot include a protected movie in a projector's play list, although you can call a protected movie using Lingo.

If you include movies in the projector play list that you jump to with Lingo, be sure to test the projector.

Projectors do not use the settings in the Director application's control panel; so, for example, if you want a projector to loop, you must use Lingo to specify looping.

Externally linked files such as digital video movies, PICTs, and sounds cannot be included in a projector. You must provide them separately along with your projector. In order to play digital video movies, users must have the QuickTime extension installed on the Macintosh.

Item	Used in a projector	Can be protected	Cross platform
Director movie	√	√	√
Shared cast	√	√	√
Protected movie	-	-	√
Linked files	-	-	√ <sup>1</sup>
QuickTime movie	-	-	√ <sup>2</sup>

<sup>1</sup> The linked files must use valid DOS filenames. When you first open the movie in Director for Windows, Director will ask you to locate the external files. When you save the movie, Director remembers the new pathnames to the linked files.

<sup>2</sup> Use the QuickTime Movie Converter to make a QuickTime movie self-contained and playable on Windows and Macintosh computers.

For more information on related topics, see:

- ◆ “Creating a shared cast” in Chapter 1 of *Using Director*
- ◆ “Movies within Director movies” in Chapter 2 of *Using Director*
- ◆ “Create Projector” in Chapter 6 of *Using Director*
- ◆ `pathName` function in the *Lingo Dictionary*.

### *Can I protect a shared cast movie or use it in a projector?*

You can protect a shared cast movie or include it in a projector. To include a shared cast movie that is not in the same folder as the projector, you must use the `pathName` Lingo function.

### *How do I set the default palette for a movie?*

Use the Movie Info command on the File menu to set the movie's default palette. Director uses the default palette until it encounters a different palette setting in the palette channel.

### *Can I use an alias for a shared cast movie?*

Director cannot load a shared cast movie using an alias named SHARED.DIR. However, you can use aliases in many other places within Director, such as:

- ◆ When you import a linked cast member
- ◆ In the go to movie, sound playFile, openXlib, and closeXlib Lingo commands
- ◆ For the movie specified using the Lingo window keyword
- ◆ For applications used by the Launch External Editor command.

### *How can I extract only the filename from a Lingo string that returns a full pathname?*

The following handler provides an example of how to strip out just the file name from the full path name in a Lingo string.

```
on textStripPathOut psString
-- simple function to strip the path out of a string and
return only the filename
-- initialize
  set lsReturnString = ""
-- store the current delimiter
  set lcStoredItemDelimiter = the itemDelimiter
  set the itemDelimiter = ":"
  set lsReturnString = the last item of psString
-- now restore current itemDelimiter
  set the itemDelimiter = lcStoredItemDelimiter
  debugMessage "textStripPathOut complete, " & lsReturnString
  return lsReturnString
end textStripPathOut
```

## *How can I make my buttons react when the user points to them or clicks them?*

The following frame script and handler provides an example of how you can simulate common button behavior. The handler can be called from a frame script, where a different action is associated with each sprite. These actions are scripted as a single string.

In this example, each button has three pieces of artwork associated with it, in three successive cast member slots. The first piece of artwork is the default button. The next piece of artwork is the highlighted button (for when the user points at, or “rolls over” the button), and last piece of artwork is the clicked (pressed) button.

Here is the sample frame script:

```
if rollover(1) then FlashRoller 1, "go to frame 10"  
if rollover(2) then FlashRoller 2, "play movie " &QUOTE&-  
  "My Drive:More Movies:NewMovie" &QUOTE  
if rollover(4) then FlashRoller 4, "set the stageColor to-  
random(256)" &RETURN& "updateStage" &RETURN&-  
"MySpecialHandler"  
go the frame
```

On the following page is the sample handler, defined in the movie script.

This handler includes extensive testing for actions in which the user points at a button or clicks a button:

```
on FlashRoller whichChannel, scriptToExecute
  set myCast = the mouseCast
  set flag = 0
  set the castNum of sprite whichChannel = myCast + 1
  updateStage
  repeat while rollover(whichChannel)
    if the mouseDown then
      set the castNum of sprite whichChannel -=
      myCast + 2
      updateStage
      repeat while the mouseDown
        if not rollover(whichChannel) then exit repeat
      end repeat
      if rollover(whichChannel) then set flag = 1
      exit repeat
    end if
  end repeat
  set the castNum of sprite whichChannel = myCast
  updateStage
  if flag then do scriptToExecute
end FlashRoller
```

# *Acknowledgments*

Written by Robin Foster, based on work by Joe Schmitz.

Edited by Diana Wynne.

Special thanks to Lalit Balchandani, Jantje Boichel, Mark Castle, John Dowdell, Jeff Essex, Steve Pasos, Dan Sadowski, Fred Sharples, and Dave Shields.