

HOUR 17



Using Video

Before Flash MX, I would get tired of telling people “No, you can’t do video in Flash.” Sure there were some creative workarounds. But no more! Flash MX lets you import video and compress it using a state-of-the-art codec that keeps files small and quality high. Even though most of the technical details are handled automatically, you’re given enough control to vary the results greatly.

In this hour, you will:

- Import and use video in your movies
- Build scripted controls to play videos
- Optimize for high quality and low file size
- Enhance quality and performance using Sorenson Squeeze with Spark Pro

NEW TERM

A *codec* is short for compressor/decompressor. Basically, videos need to be compressed so that they download quickly. The same video needs to be decompressed for the user to view it. Many different technologies are available to compress and decompress. The codec that Flash uses is called Sorenson Spark.

Importing Video

Importing video into Flash is only slightly more involved than importing other media types such as sound or graphics. It really is as easy as selecting File, Import and selecting a video. But just like sound and graphic files, there are different types of video files—each with its respective attributes. In addition, the tasks of making a video look good, play well, and download quickly can all be involved. The first step—always—is concentrating on your source video.

Making Video Look Good

We'll look at specific format types in a minute, but it's worth noting that the entire creation process for video involves many specialties. Even if you don't employ a Hollywood crew, you will always have the responsibilities of sound engineer, camera operator, casting agent, writer, and so on. Also, technical issues such as background noise, camera shake, and lighting all affect the final outcome. There's not enough room in this book to teach you everything about video making. Just consider learning about traditional movie making.

When selecting or creating videos, there are a few considerations specific to video on the computer. The entire “viewing experience” is much different on a computer than in your living room. The following tips come down to the fact that video on the computer has restrictions. (Additionally, there are other issues related to compression discussed later this hour.)

As far as video content is concerned, refrain from using many fine details such as small text—it can become illegible in a small video window. Changing viewpoints and camera angles (such as going from close up to far away) is always good—just be aware of details that may be difficult to see on a small screen. Also be conservative with special effects; everything has a “price”—be it lower performance, large files that download slowly, or gratuitous effects that become tiresome. Although the limits of computer videos may appear restrictive, they are really just challenges to overcome. In some ways, the restrictions of regular TV are removed. For example, you can use nearly any aspect ratio (that is, the shape—wide or narrow). You can even mask the video to make any shape you want as Figure 17.1 shows.

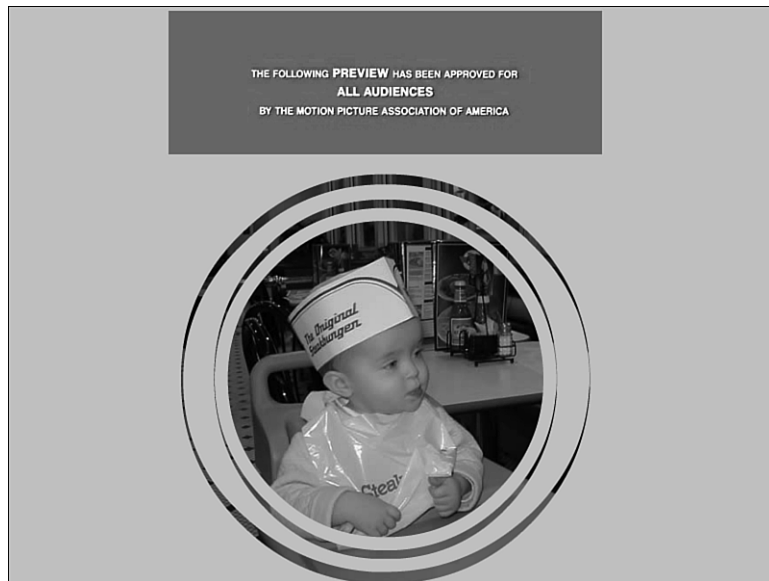
Supported Formats

Flash supports video in that you can import source digital video footage, compress it (so that the files aren't huge), use the video in your Flash movie, and then export as a .swf

file to post on your Web site. Notice that you always must start with a source video. In addition, it must be a digital file saved on your computer. For example, if all you have is a videotape, you'll need to digitize it first. This involves video capture hardware, which basically records the analog video into the computer's hard disk. However, if your source video is already in a digital format (such as a mini DV video camera), making a digital copy on the computer takes little more than a cord to connect the camera to the computer (naturally, this is also a piece of hardware). Finally, although your favorite movie on DVD is in fact already saved as a digital file, you'll find it next to impossible to convert this to a digital file that Flash will import (besides the fact that you'd likely have to break serious copyright laws).

FIGURE 17.1

Despite computer video's restrictions, you can do cool stuff such as change the viewing window's shape.



17

There's much more to say about the way you create high-quality digital video files on the computer. However you do it, you must create one of the following digital file formats:

- QuickTime (.mov)
- Digital Video (.dv or .dvi)
- MPEG (.mpg or .mpeg)
- Windows Media (.asf or .wmv)
- Video for Windows (.avi)



It's possible that your system configuration won't list this many digital video formats. To increase the types of video your computer will support, install QuickTime 4 or later (from www.quicktime.com). Additionally, if you're on Windows, be sure to install DirectX 7 or later (www.microsoft.com/directx/). Direct X 8.1 is preinstalled on Windows XP.

Although the list of video formats is relatively long, you'll probably only use one of the first three. Although each format has its unique purpose, when it comes to importing video into Flash, you always want to start with the highest quality source. That's because Flash always compresses the video (a little or a lot—but always some). Video is like images in that quality degrades when you compress a file that's already compressed. Unfortunately, true uncompressed video files will be huge. In practice, a video that's only slightly compressed will still be high in quality but not nearly as big.

Professionals creating source digital video files should have no trouble creating a QuickTime with little or no compression. Also, if you're copying video directly from a digital video camera, you can use one of the DV formats. Many cameras will let you save a QuickTime video alternatively. Although MPEG can be high quality, there are actually several versions, so you'll have to test your entire system to make sure that it works for your needs. I suspect that many MPEGs you'll find have already been compressed, so you really should find a higher quality original. The same can be said for Windows Media. I think you'll be hard-pressed to find non-compressed high-quality .asf or .wmv files. Finally, Video for Windows (.avi) files are simply not as good as QuickTime. They're old technology. (This isn't to say that you can't make a decent .avi, but just that you'll do better with QuickTime.)

Don't let this section sound like a rule book for how you should progress. Basically, you just want to start with the highest quality original.



If you haven't already, you should definitely download and install QuickTime from www.apple.com/quicktime/download/ (for the exercises in this book, you can use the free player). QuickTime is arguably the best product Apple has ever created. The fact QuickTime offers such high-quality digital video (for both Macintosh and Windows) is the reason that most new movie previews are available in QuickTime.

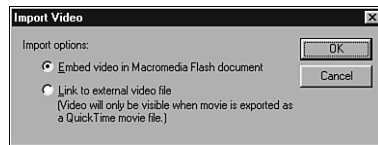
Task: Import a Video

In this task, you will import a QuickTime video into your Flash movie. Here are the steps to follow:

1. To make this go quickly, we'll just use the sample movie that comes with QuickTime. Start a new Flash movie and select File, Import. Select the file sample.mov, which is located in the QuickTime folder (inside Program Files/QuickTime).
2. The first of three dialog boxes that appear asks whether you want to link or embed the video (see Figure 17.2). You only see this option when importing the QuickTime video format because linking to an external video works only with QuickTime. More important, linking is an option only when you plan to distribute a QuickTime video—not an .swf file. This is slightly confusing because Flash can produce QuickTime videos that only play in the QuickTime player. We'll touch on this again in Hour 24, “Publishing Your Creation,” (because, after all, QuickTime is a publishing option). Anyway, for this exercise, select the embed option.

FIGURE 17.2

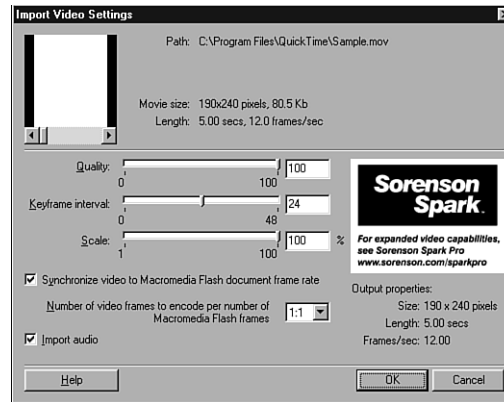
When importing QuickTime, the option to link (not embed) is used only when you plan to export QuickTime videos.



3. The next dialog box that appears includes Flash's built-in settings for Sorenson Spark compression (a proprietary technology included with Flash). The selections you make in this dialog box affect the ensuing process that's about to occur. The process is select a video to import, choose the compression settings, let Flash compress the video, and then use the video in your Flash timeline. The settings you make here are comparable to the settings you make for imported raster images; however, in this case, compression happens first, and you can't change the settings (where in the case of graphics you can keep clicking the Library's Test button after every change). After all this talk, just leave all the defaults—make sure that Quality is 100 and that the Synchronize Video... check box is selected as in Figure 17.3; then click OK.
4. The final dialog box appears to ask whether you want to extend the current timeline to 61 frames (the length of the video). Click Yes.

FIGURE 17.3

The choices for Sorenson Spark compression appear once upon importing a video.



5. At this point, you'll be back in Flash with the movie you just imported on the Stage. (The first frame of this particular movie is white, so you might move the red current frame marker out past frame 20 or so.)
6. That's about it! You need to test the movie to hear the audio play.

If that's all you wanted (a video inside Flash), you could stop there. You could use the Info or Align panels to move the movie to the top-left corner and then set the document properties to the same size as the video. Just click the video to see its coordinates in the Properties panel; then click the Stage, click the Size button, and change the document properties to match the movie size (190x240 in this case). However, if all you wanted was a video inside Flash with nothing else, I'd suggest you consider simply using QuickTime instead.

Anyway, I hope you want to do more than just watch a video inside Flash. So much more is possible, and you'll see some of it next.

Using Video

You need to understand one primary concept when using video in Flash: sounds perform like stream sounds, and pictures act like Graphic symbols. This isn't "streaming" where you can experience the beginning of a media source while the rest downloads (although Flash does this automatically). Recall that sounds in Flash can be set to Stream or Event, and it makes a difference. Additionally, remember how an instance of a Graphic symbol differs from one of a Movie Clip. (These topics were covered in Hour 10, "Including Sound in Your Animation," and Hour 12, "Animating Using Movie Clip and Graphic Symbols," respectively.)

As a review, a sound in a keyframe set to Event begins to play when that frame is reached and continues to play until it's finished—even if the timeline slows down to display all the frames. Stream sounds are locked to the timeline. This means that you need to make sure that the sound has enough space in the timeline to finish. The reason to review Stream sounds is that a video placed in a keyframe needs enough space in the timeline to finish playing the audio portion. For example, if the video is 10 seconds long and the frame rate is 12 fps, you need 120 frames in the timeline. Another reason to remember Stream is that by default all the audio in your video is affected by the global publish setting for stream audio (File, Publish Settings, Flash tab).

The visuals in your video behave like a Graphic symbol. In fact, videos are not really the same as Graphics, Buttons, or Movie Clips because you can't tween videos or tint them (through the color effects like other symbols). However, videos are most like Graphics in several respects. You can scrub to see a preview of videos. Also, you must extend the timeline far enough for all the video frames to play (like you did in step 4 of the preceding task).

Really, these concepts are just here to help you understand a few of the technical details that follow. Also, don't worry if the fact that you can't tween videos or tint them scares you. You'll see in the next task that after you place a video inside a Movie Clip, you're given all the controls you'll ever need.

Task: Make a Playback Controller for Video

In this task, you'll add some standard video buttons that give the user a way to control the video. Follow these steps:

1. Create a new movie and import a video. You can use the same sample movie from the preceding task. The only difference this time is click No on the final dialog box (the one that asks you whether you want to extend the timeline—see Figure 17.4).

FIGURE 17.4

For this exercise, don't let Flash automatically insert extra frames immediately after import.



2. Now that the movie is embedded and compressed, we can put it inside a Movie Clip. Select the video on the Stage and convert to symbol (press F8). Select Movie Clip in the dialog box and call this symbol myVideo.

3. Immediately after you create the symbol, Flash asks again whether you want to extend the timeline—this time the Movie Clip’s timeline. Click Yes.
4. You’re about to create buttons (for Stop, Pause, and Play). Just to make it easy, we’ll put those buttons inside the myVideo symbol. Double-click on the myVideo symbol on the Stage to be taken inside the master symbol. (If you wanted the buttons outside the symbol, you would need to give the myVideo symbol an instance name and precede each of the following lines of code with that name plus a dot.)
5. While inside the master myVideo symbol, insert a new layer for the button. Select Window, Common Libraries, Buttons. Inside this library, you’ll find a folder called Playback. Drag on each of the following buttons: gelPause, gelRight, gelStop. You can align these underneath the video however you want (consider using the Align panel).
6. Open the Actions panel and for each button, click once to select it; then insert an on mouse event (from the plus button, select Actions, Movie Control). (You can keep the default “release” option.)
7. For each button, you want an additional script placed between the curly braces of the on mouse event. Specifically, for the gelRight button insert a `play()` action (Actions, Movie Control). For the gelPause button, insert `stop()`. And, for gelStop, insert a `goto` and then change the option to go to and stop. All three buttons’ actions are shown in Figure 17.5. (If you think I made a typo, think about it for a second—the right arrow is definitely `play()`; the Pause button is really `stop()` because Flash has no “pause” action; and the Stop button is really “go back to the beginning and stop.”)
8. Finally, insert another layer in the myVideo symbol and then insert a keyframe in the last frame. Place a `stop()` action there.
9. Do a test movie, and you should be able to control your imported video.

Advanced Video Controls

In the preceding task, you saw how easy it is to include simple controls. It’s not much more difficult to add advanced features. For example, having both a Pause and a Play button onscreen at the same time is a little weird. After all, why would you need to press Pause if the video is already stopped? One solution is to create another Movie Clip (nested inside the myVideo symbol) that contains both the Pause and Play buttons but in separate frames. That is, in frame 1 place the Pause button and then in frame 2 (and whole new keyframe) place the Play button. Then you need to edit the script attached to each button. The Pause button’s `stop()` Action needs to change to `_parent.stop()` because it is the timeline up the hierarchy that needs to stop. Similarly, the Play button’s

`play()` needs to change to `_parent.play()`. Finally, both buttons need an additional line of code: `gotoAndStop(2)` for the Pause button and `gotoAndStop(1)` for the Play button. That's so that the clip (containing the buttons) will jump to the other frame. Finally, this button clip needs a `stop()` on the first keyframe so that the Pause button appears at the start (and so that you don't see Play and Pause flickering as the movie clip loops by default—see Figure 17.6).

FIGURE 17.5

The final code for each button.

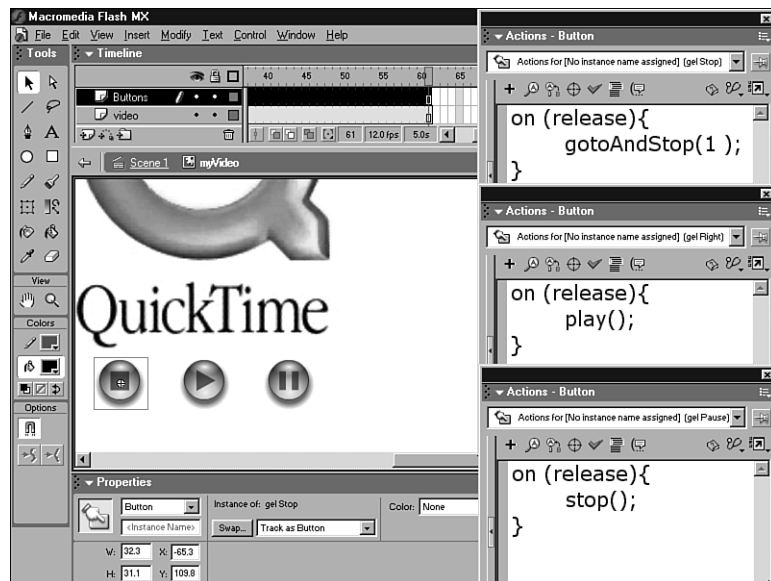
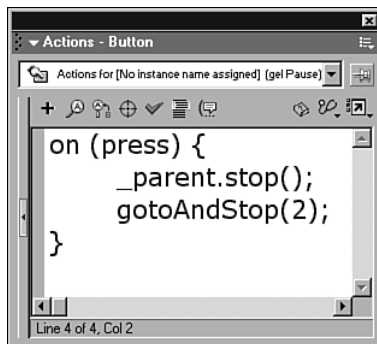


FIGURE 17.6

The modified code to make a pause/play toggle clip.



There's still more you can do! Consider a progress bar that shows you how much of the video has played. A quick-and-dirty way to make one is to first create another layer

above the video. Draw a rectangle that becomes the progress bar. Convert this to a symbol (as a Motion tween will be perfect) but select the middle left default center option in the Convert to Symbol dialog box shown in Figure 17.7. This way, the rectangle can have its horizontal axis scaled way down and then appear to grow to the right (during the tween). That is, just add a keyframe at the end of the progress bar's layer and then transform the rectangle in frame 1 by grabbing the middle horizontal handle on the right and scaling down to the left (see Figure 17.8). Then just set the first keyframe to motion tween, and you've got a progress bar. Add still another layer for an outline showing the progress bar's maximum width, and it looks pretty good.

FIGURE 17.7

Selecting the middle left default center point will make the progress bar's scale appear to grow to the right during a tween.

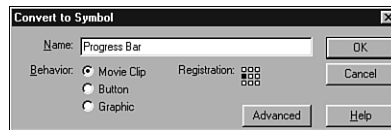
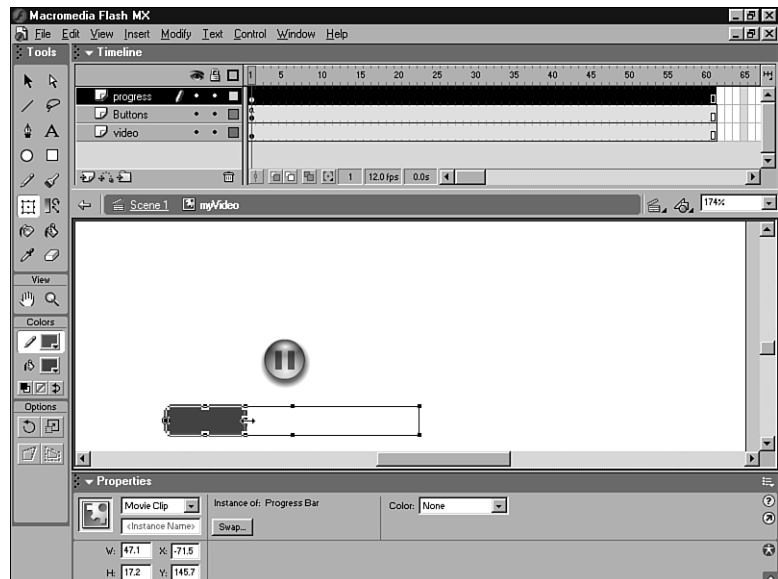


FIGURE 17.8

Transforming the scale of the progress bar involves grabbing the middle right handle.



You can probably think of more effects that could be appropriate for your movies. You can easily add mask layers or even tween the clip containing the video itself. Ultimately, however, videos can add a lot to your file size. For this reason, make sure that you learn

about the `loadMovie()` action covered in Hour 19, “Designing Your Web Site to be Modular.” Also, Hour 20, “Optimizing Your Flash Site,” teaches you the skills necessary to analyze and optimize your movies. You’re about to learn optimization tips for video, but realize that even more general techniques can also be applied.

Optimizing Video

Two topics relate to optimization: optimizing performance and optimizing download speed. Although these are, indeed, two different topics, in the case of video, they’re often related. That is, a video file with many detailed frames tends to be large (so it downloads slower), and it also tends to perform worse because there’s so much data that needs to be uncompressed and displayed onscreen. The topic of playback performance is discussed in greater detail later in Hour 20. For now, let’s talk about how to make video files small.

It may seem obvious that the reason to keep Flash movies small is to minimize download time. However, because Flash can stream (meaning that the movie starts playing before it’s entirely downloaded), you don’t need to worry so much about the entire movie’s download time, rather just the pause before playback begins. For example, if a movie takes 2 minutes to download, that’s only “long” if the movie’s playback time is 10 seconds. Conversely, a 2-minute download for a movie that plays for 2 minutes isn’t long at all considering it will start playing after only a 12-second pause (later frames are downloaded while the user watches the first frames). I calculated the preload using the following formula:

$$\text{preload} = (\text{download time}) \times (\text{play time}) + (10\% \text{ of play time})$$

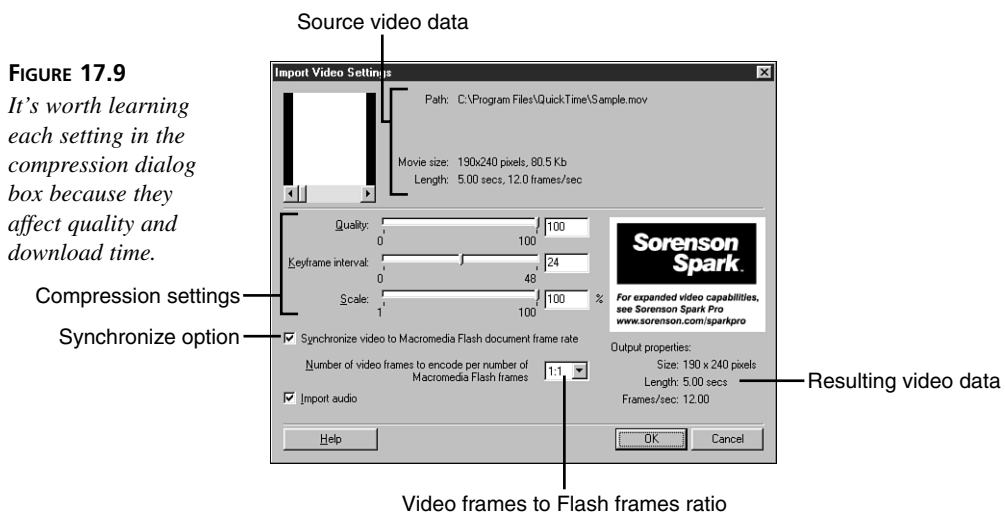
So really, the concern is keeping that preload time to a minimum. The preload time is a factor of the relationship between download time and play time (plus a 10% safety for fluctuations in connection speed). It just so happens that a video with a long playback time is nearly always going to have a similarly large download time—but not always. The point is that we care about the preload, not necessarily the total download time. By the way, the Bandwidth Profiler (introduced in Hour 20) will be instrumental for calculating actual download and preload times.

Settings for Sorenson Spark

Every time you import video into Flash you’re given options as to how you want to compress the video. The compression technology used in Flash is called Spark and is licensed from Sorenson. During the process of compression, your video is analyzed for ways that similar data (say the portion of an unmoving background) can be recycled to make the file smaller. Basically, the video is compressed to make the file smaller (not

unlike “zipping” or “stuffing” a file). Then after it’s downloaded and playing on the user’s machine, it’s decompressed. In the case of zipping or stuffing a file, you want all the data in the original to be restored. That’s called *lossless compression*. In the case of video, you want to use so-called *lossy compression* because it makes the file even smaller. The only side effect is that data is lost. Just like .JPG images or .MP3 sounds (which are also lossy) that’s not a problem because the data that’s lost is usually not noticeable. It just depends; when you compress a lot, the quality becomes notably worse.

The simple rule for compression is: compress as much as possible without compressing so much that quality becomes unacceptable. Obviously, this is subjective. And, unfortunately, video compression can be complex because multiple images as well as sound all factor into the process. Luckily, you don’t have to learn any more than the few dials that appear in the Spark compression dialog box shown in Figure 17.9.



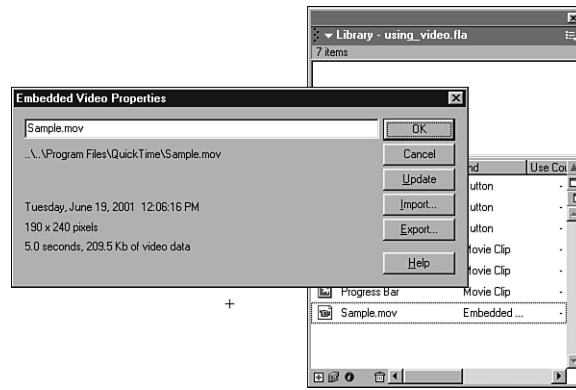
You need to make several considerations when configuring the Spark compression dialog box. Probably the first thing to note is the source video’s data. The Movie Size includes the pixel dimensions (190×240 in Figure 17.9). If you want, you can scale that down by dragging the Scale slider or by typing a percentage. Any change you make will be reflected in the output properties on the right. Without exception, a scaled-down video (besides looking smaller) results in a smaller file size.

The Quality setting is the most difficult to judge because you won’t see the results of a higher or lower quality until after you click OK. If you’re unhappy with the quality results (either it doesn’t look good or you think you could sacrifice even more quality),

you'll need to re-import the video to make another selection. I suggest always setting that at least a tad lower than 100—even just down to 99 will result in a smaller file without noticeable quality degradation. Actually, the best process is to go “too far” toward low quality and then back off (going higher) until you're satisfied. (Again, you'll have to re-import the video to make new compression settings—but that's easily done from the Library item's properties dialog box shown in Figure 17.10).

FIGURE 17.10

To re-import (and choose different compression settings), select the properties option for the Library item.



Although Scale and Quality are subjective decisions, they happen to be the easiest to understand. Keyframe Interval requires you know a little about how Sorenson Spark compression works. It's called “temporal” (or over time), meaning that the entire first frame of the video is saved. And then, during the ensuing frames, only the difference that appears over time is saved (which saves space) until the following keyframe where the whole frame is saved again.

The Keyframe Interval lets you specify how far apart the keyframes occur. That is, exactly how many “interframes” before the next keyframe. The simple rule is a lower Keyframe Interval means more keyframes, which means a larger file but one that allows for better instant-access to any one frame. You should only select a low Keyframe Interval if you plan to script a way to play the video backward, or if the visual content is fast changing. Similarly, only select a high Keyframe Interval if the visual content is slow changing. Ultimately, you just have to play around with it and note how performance and file size are affected.

Finally, the synchronization options let you control how a video with one frame rate will play inside Flash, which has its own frame rate (which may or may not be the same). Take the likely situation in which your original video's frame rate is 30 fps, but you plan to keep your Flash movie at a more conservative 12 fps. If the video is 10 seconds long,

that's a total of 300 frames. But a Flash movie at 12 fps will only display 120 frames in the same 10 seconds.

There are many options for how to deal with this mismatch. For example, you may want to import only 12 frames for every second of video; in which case, you'd only import fewer than every other frame. Or, you might want to import all 300 frames, but then the video would appear to play slowly (in addition to having audio drift out of synch—and, no, you can't just play the audio slowly). You'll usually just select the option Synchronize Video to Macromedia Flash Document Frame Rate, which translates to mean: I want the video to match my current frame rate (in Flash) even if that means frames are skipped (called *drop frames*) during import. Realize, of course, that the motion in a 30 fps video playing at 12 fps (and having dropped frames) will look jumpier than it will at 30 fps.

In some situations you'll want to modify the default synchronize option or even turn it off. If you're importing video originally created for television, the video's frame rate may likely be the odd number 29.97 fps. In fact, this is close enough to 30 fps, where if the audio were to drift out of sync it would take a long time to notice. However, if your Flash movie is at 30 fps and you select the synchronize option, you may see a subtle visual artifact where the graphics jitter periodically. It's definitely subtle, and I can't say that I've actually seen it—but it's a real thing.

Usually, original videos will have a higher frame rate, in which case, you'll have to decide whether to drop frames to make it match. However, it's possible that the video was created at a lower frame rate. If you select the synchronize option in this case, you'll actually add paused frames to make the video match. Normally, this is not a problem; however, it's worth considering lowering the Flash movie's frame rate (that tends to make everything smaller because there will be fewer total frames). By the way, if you don't synchronize in this case (of a video with a lower frame rate), the video will play extra fast.

Just to be complete (and to perhaps add more detail than you want), I should mention a setting called Video Frames to Flash Frames Ratio. We already considered the case of a video with a higher frame rate than Flash and how you could select the synchronize option to drop the appropriate number of frames to make it stay synchronized. But what if you want to drop even more frames to make the file smaller? Naturally, the motion in the video will look even worse—but it may be worth it. The drop-down list normally shows 1:1, meaning that one video frame will be compressed for every one frame in Flash. If you set it to 1:2 that means only one video frame will be compressed for every two frames in Flash. This forces every other frame that would have otherwise been compressed to be dropped.

You can select other ratios, and each one means that the video will be jumpier but smaller. I understood this option much better when I explored different options and simply looked at the resulting output properties on the dialog box. Finally, realize that this ratio option can be independent of the synchronize. Think of it this way: Synchronize has priority, meaning that necessary adjustments are made for this first. Then, if you select a ratio other than 1:1, Flash looks at the frames it planned on compressing and then compresses even fewer depending on your setting.

Here's a quick rundown of my compression suggestions:

- Select the scale (or, really, the final window size) first.
- Determine the target file size (based on how much of a pause the viewer will endure).
- Determine the Flash movie's frame rate before importing a video because this affects the synchronization options.
- Think about the nature of your content. Does image quality or motion smoothness have a higher priority? Tinker with both the quality setting and the video-to-Flash frames ratio (to drop frames).
- Always lower the quality a little—even if just to 99.
- Don't mess with the keyframe interval unless the user needs quick random access to any specific frame (in which case, you lower the keyframe interval).

Other Video Options

Believe it or not, there's so much more about video that a whole book would probably be too little. But even if I restrict the topic to video in Flash, there are a couple other points to make. You've already learned how to import a video and compress it (immediately after import). This process was relatively time-consuming and somewhat limited. Compressing the sample movie went pretty fast, but it was only 5 seconds long. You'll see much longer compression times for longer movies. And, despite the fact that there are many options in the Sorenson Spark dialog box, many more controls are possible with the Professional version of Spark. Next we'll look at how you can reduce that compression time and then we'll look at the third-party product Sorenson Squeeze with Spark Pro.

Offline Preparation

Flash can import several different video formats (listed earlier this hour). For each format listed, you must compress it individually at the time of import. This can be a significant

investment of time and capitalization of your computer. One other video format that I didn't list is also supported by Flash. The file format with the extension .flv was created just for Flash (they're called Flash Video files). Basically, they're nothing more than videos that have already been compressed using Sorenson Spark. The great part is that they import almost instantly!

You can create .flv files in two ways. One way is to import original videos and make selections in the Spark dialog box (just as you did earlier). Then you can export an .flv file from Flash. The other way is to use another product called Sorenson Squeeze with Spark Pro. You'll learn more about this product in a minute. First let me show you how to export an .flv file from Flash.

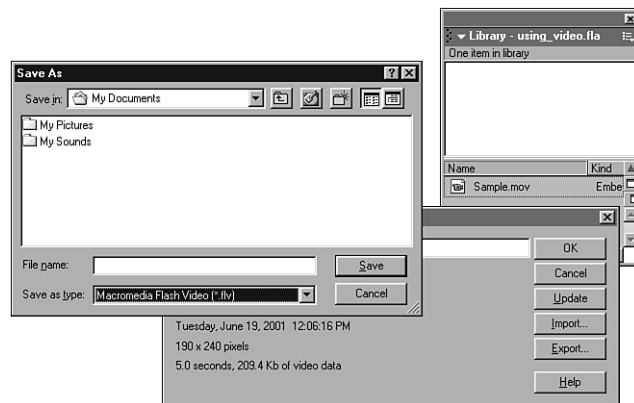
Task: Export an .flv File

These are the steps to export an (fast importing) .flv file from Flash:

1. Either open one of the files from an earlier task this hour or import a video into a new Flash movie.
2. Select the video item in the Library and select Properties from the Library's Options menu.
3. Click Export and then save the .flv file in a memorable place (say, the desktop—see Figure 17.11).

FIGURE 17.11

Exporting an .flv file is as easy as clicking a button.



4. That's it! The .flv file contains all the compression information you selected on import. Actually, it's already compressed.
5. Start a new file; then select File, Import and point to the .flv file you just exported. Notice this time that it takes only an instant to import and doesn't require any compression settings.

Although it may seem like this task was easy (it was), you may wonder why you'd want to bother going through these extra steps. After all, you have to sit through compressing the video in any case. This process can make you more productive. For example, one person could spend all day importing source videos and compressing them, while others work on other things. Even if you're the one who has to do everything, you could spend concentrated time doing the compression. Consider that you'll often import and compress a video and then decide that the results are not ideal—so you try again. You can use one work file into which you import videos and promptly export them as .flv files. Then those .flv files can be imported into the final files.

Using Squeeze with Spark Pro

There's one other way to create those fast-importing .flv files. It's a moderately priced product called Sorenson Squeeze with Spark Pro. This product produces .flv files, but it also happens to export QuickTime videos and other formats. There are two main features that, if you use them, could justify the product's cost. First, it can batch process many files. That means that if you have 100 videos to compress, you can set up the batch and then leave for as many hours as necessary while the computer processes each video one at a time.

The other feature of Squeeze is that the options for compression are much more detailed. Think of it this way: Spark (the technology included in Flash) has the main features needed to compress videos. Spark Pro (included in Squeeze) has extra fine-tuning features and uses technology not available in the regular version of Spark. In any case, after the video is in Flash, any user with the Flash version 6 player will be able to view the movie.

To get an idea of the additional options available, take a look at the filter settings in Figure 17.12.

You can also see that although the adjustments are similar to those inside Flash, you do get more fine-tuning capabilities. For example, you can specify your target download data rate before compressing. These options (shown in Figure 17.13) can also be saved in sets so that you can have one combination for modem connections and another for broadband connections.

Finally, there are some particularly interesting technologies available only in Spark Pro. One is called Sorenson 2 pass Variable Bit Rate, which basically means that the data rate varies as needed throughout the video—saving file size when less movement is onscreen for example. There's another interesting option that let's you compress audio with Fraunhofer MP3 (which is really just an additional choice for how the audio is compressed). However, because all the audio in a video is compressed using the settings you

make in Flash's Publish Settings dialog box this setting gets overridden when you use Squeeze to create an .flv file. (You can get around this problem by exporting .swf files from Squeeze and then using Flash's `loadMovie()` command to import at runtime.)

FIGURE 17.12

The filters let you clean up a video, crop, or add a fade.

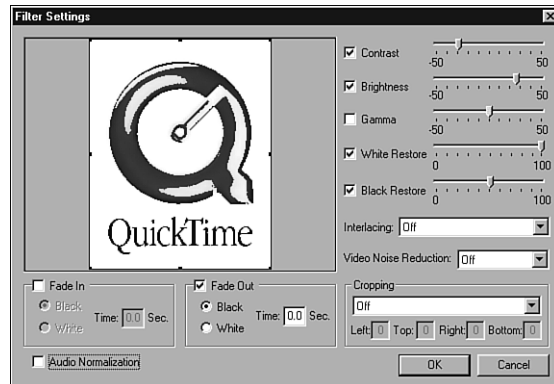
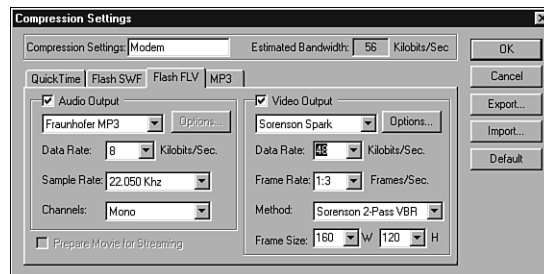


FIGURE 17.13

Some of the additional compression options available in Squeeze.



Hopefully, this section doesn't sound too much like a commercial—I don't want it to. However, Sorenson compression is widely accepted as the de facto standard for digital video. If you get serious about video in Flash, you should consider all the options available.

Summary

Step back and remember the definition of video: synchronized images and sound. Some messages are simply best suited for video. In my mind, only video reveals the personality of a subject. Also, when a demonstration is needed, there's just nothing like a video. Because the majority of this chapter dealt with the limiting factors for video, don't be afraid to use video when appropriate. The "price" (extra work on your part and extra download time for the user) can definitely be worth it when necessary.

When you import a source digital video file, Flash uses Spark compression to make the file smaller. You can make the file even smaller (but usually with lower quality) by changing the compression settings. The Spark decompression technology is used in the Flash player when the user watches your movie.

Using the video in your movie is slightly restrictive because you need to treat the video like a stream sound and graphic symbol. That is, a video clip needs space in the timeline (depending on the video's duration).

This hour, you got to see all the standard compression settings available in Flash. Also, you got a quick look at Sorenson Squeeze with Spark Pro that gives you even more compression fine-tuning.

Q&A

17

Q I realize that you recommend against using .avi files, but that's all I have. Should I first convert it to a QuickTime (using a tool such as the QuickTime player)?

A First, if the only source file is an .avi you might as well use that. If it's good quality, great. If it's not, oh well. But, converting it to a QuickTime can't make it better.

Q I've imported a video with a great music track. Why then, when I select minimum levels of compression does the audio sound so terrible?

A You need to set the Audio Stream compression and quality from Flash's Publish Settings (Flash tab).

Q I know that the compression stage can take a long time, but even after I've compressed the video my Flash movie takes forever to export. Why is that?

A That's because audio takes a long time to compress, and it's likely that Flash is compressing the audio at the time you publish. You can temporarily change the Audio Stream setting (in Flash's Publish Settings dialog box) to Raw so that every time you do a test movie it goes quicker. Just remember to set it back to a reasonable compression level before you export the final time.

Q I've imported and compressed a QuickTime that I downloaded, but my Flash movie is larger than the original video—how can that be?

A Generally, this won't happen. However, if the video was compressed before you imported it, it's possible that it actually grows in size. Again, try to start with the best quality uncompressed original. Also, consider just leaving the video as a QuickTime and linking to it.

Workshop

The quiz and exercise questions are designed to test your knowledge of the material covered in this hour.

Quiz

1. If your video has a frame rate of 29.97 fps (a likely rate if it was made for television), the videos in your Flash movie will be slightly out of sync with the audio, unless you do what:
 - A. Use video from Europe on the PAL system (which is exactly 25 fps).
 - B. Click the Synchronize option when importing.
 - C. Use 30 fps in your Flash movie because this is “close enough for television.”
2. How do you change the compression on a video that you’ve already imported?
 - A. You can’t. But, instead, you could re-import and recompress at that time.
 - B. Simply access the video item (in the Library), select its properties option, and then press recompress.
 - C. You simply need to modify the Publish Settings, Video tab.
3. Sorenson Spark is...
 - A. An old technology that could cause sparks to shoot from your monitor.
 - B. The compression technology included free with Flash MX and the decompression technology used in the Flash player.
 - C. An option (that costs extra) to compress your videos with supercompression.

Quiz Answers

1. B. It’s that easy. In fact, it’s probably not terribly noticeable. Also, PAL video is 25 fps, but that has nothing to do with the question.
2. A. Answer B is actually close because you can re-import from the item’s properties—but that just invokes the compression dialog box again (and requires that the original video file still be in its original location).
3. B. As a “codec,” there are both compression and decompression components. There is a version called Spark Pro (that costs extra), but realize that users with the Flash player will still be able to view Flash videos that use this version.

Exercises

Go ahead and try to make a toggle play/pause and a progress bar (as described earlier in the “Advanced Video Controls” section). I’ve posted a complete file at www.phillipkerman.com/teachyourself/sourcefiles so that you can see exactly what I had in mind.