22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 387

HOUR 18

Using Video

What You'll Learn in This Hour:

- How to import and use video in movies
- How to build scripted controls to play videos
- How to optimize for high quality and low file size
- How to use cue points to synchronize a video to the graphics in your Flash movie

In the not-too-distant past, it was impossible to display video inside of Flash without something short of a miracle. The video support in Flash 8 has really come of age. You can import video, trim it, and compress it, using the state-of-the-art VP6 codec from On2 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. If you want the best quality and most features, you need to use Flash Professional 8. There are a few workarounds that you can take if you only have Flash Basic 8 but, in short, Flash Basic 8 falls flat if you want to do video.

Codec is short for *compressor/decompressor*. Basically, videos need to be compressed so 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 Basic uses is called Spark (from Sorenson) while Flash Professional 8 also includes VP6 (from On2).

Flash Basic 8 Versus Flash Professional 8

You're about to learn nearly every option available for preparing and displaying videos inside Flash. Because the differences between Flash Basic 8 and Flash Professional 8 are most striking when considering video, I've produced the following table as an overview.

22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 388

Don't worry if you don't understand every technical term—you will by the end of this hour. You can use the following table as a reference as you go through the rest of this hour.

Technique	Flash Basic 8	Flash Professional 8
Embedding a video into your `.swf` file	Supported.	Supported.
Encoding a video as an `.flv` file for external playback	Workaround: embed it first, then export. Note, these `.flv`s are CBR, not VBR.	Supported Also included is a standalone application for batch `.flv` creation.
Playing an external `.flv` (including those produced in outside applications)	Only with the "no-frills" video object.	Can use video object, Media Components, and the new FLVPLayback component with skin chooser.
Customized encoding settings (instead of presets)	Unavailable.	Supported.

Embedding Video Versus Playing External Video

There are two general ways to play video inside Flash: either embed where you import, compress, and then place the video inside your main file or, compress the video into an `.flv` file and play it externally. The video file has to download in either case, but by keeping it external you can control when it downloads. Perhaps you can give the user a choice of three different videos and only download the one they select. Embedding the video not only makes your main file bigger, but the quality isn't as good as playing an external `.flv`. Generally, the best route is to create an `.flv` to play externally. However there are two cases where embedding the video is a better option. One is when the video is very short (say, less than 30 seconds) and you don't want to bother with maintaining several files (the `.swf` and the `.flv`). I say short because a long embedded video with sound will begin to drift out of synchronization. The second reason to embed video is when you need frame-by-frame control. An external `.flv` can only pause on keyframes (which get added automatically, but may only appear a few times per second—much less frequently than every frame of video). If you want to let the user step frame-by-frame through a video clip of your golf swing, embedded video is a better choice.

22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 389

Regardless of whether you plan to embed the final video or play an external `.flv`, you need to compress the video. That is, your source can be one of many video formats (such as MPEG or QuickTime) but it must be compressed using one of the Flash Player's two supported codecs. Namely, the older Sorenson Spark (which works in Flash Player 7 and Flash Player 6) or the newer, and much better quality, On2 VP6 codec (which requires users have Flash Player 8 or later).

Compressing a video (for embedding or to make an `.flv`) 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 then selecting a video. But as with sound and graphic files, there are different types of video files—each with its own attributes. In addition, the tasks of making a video look good, play well, and download quickly can all require some work. The first step—always—is concentrating on your source video.

Making Video Look Good

We'll look at the supported video types and how to get them into Flash 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 you're 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:

- Only use video when appropriate. Three particular cases stand out: celebrity endorsement (that is, when the face is recognizable or important to add credibility to your message); call to action (when the video or sound can actually motivate someone to physically get up and go do what you're pushing); expert demonstrations (for example, a cook showing you how to fold eggs is something you could never really describe with pictures or words, you really need a video).
- As far as video content is concerned, you should 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.

22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 390

- 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 18.1 shows.

In addition to these tips, you need to consider other optimization issues, as discussed later this hour.

FIGURE 18.1
Despite digital video's restrictions, you can do cool stuff such as change the viewing window's shape.

Supported Formats

Flash can compress video in various file formats. These are all digital files 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 may break copyright laws to import it).

22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 391

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`)
- MPEG (`.mpg` or `.mpeg`)
- Windows Media (`.asf` or `.wmv`)
- Video for Windows (`.avi`)

By the Way

Installing Video Support

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 (from www.quicktime.com). In addition, if you're using Windows, be sure to install DirectX 9 or later (www.microsoft.com/directx/). You can see which version, if any, you already have by selecting Start, Run, and then typing `dxdiag` and click OK.

Although the list of video formats is relatively long, you'll probably use only 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 are huge. For example, the source videos I used in a recent project were about 200MB to 300MB per minute. Starting with such high quality really made a difference, even though the compressed version was less than 1MB per minute. In practice, a video that's only slightly compressed will still be high in quality but not nearly as big. Keep in mind, however, that the compression stage can be *very* time consuming. The On2 VP6 codec is particularly asymmetric meaning it takes longer to compress than decompress. That's usually the case with any codec, but here you could be tying up your machine for hours if not days!

Professionals creating source digital video files should have no trouble creating a QuickTime video with little or no compression (for example, the codec called "Animation"). Also, if you're copying video directly from a digital video camera, you can use one of the digital video formats (`.dv`). Also, many cameras will let you save a QuickTime video. Although MPEG can be high quality, there are actually several versions of it, so you'll have to test your entire process to make sure that it imports into

22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 392

Flash correctly. 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 noncompressed high-quality `.asf` or `.wmv` files. Finally, Video for Windows (`.avi`) files are simply not as good as QuickTime files. 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 think this section is a rule book for how you should progress. The only universal rule is that you always want to start with the highest-quality original.

Now that you have your computer set up with video support, you'll see how to import video into a Flash movie in the following task.

Try It Yourself

Embed a Video

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

1. To make this go quickly, I'm going to use the sample movie that came with the free version of QuickTime (which you can find at www.quicktime.com), but you can use any footage you can find. Start a new Flash movie and select File, Import, Import Video. Click the Browse button and select the file `sample.mov`, which is located in the `QuickTime` folder (that is, `Program Files/QuickTime`). Then click Next.

2. The Deployment step of the video import sequence will appear as shown in Figure 18.2. Select the Embed option so that the compressed video will get saved inside the `.fla` and then click Next. (In fact, Flash Basic only has two options, but I'll cover all 5 options available to Flash Professional 8 after this task.)

3. The next step, Embedding, lets you select various details about how the video will appear in your file. Most of these settings are purely for convenience. For example, I recommend simply leaving the default symbol type "Embedded video" because it's simple to make the other types (Movie Clip or Graphic) by hand. For this exercise, leave the Symbol type and Audio track options as is and uncheck the two checkboxes. The only setting here that has any lasting impact is when you want to chop up your video. We'll take a quick side trip to see the video split feature so click "Edit the video first" and press Next.

22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 393

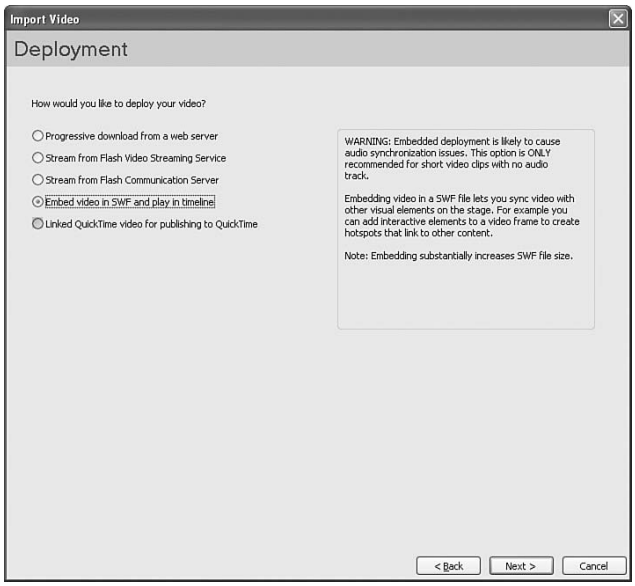

FIGURE 18.2
The Deployment dialog is where you decide whether to embed or create an external `.flv`.

4. The Split video dialog appears as shown in Figure 18.3. This is perhaps the simplest video editor of all time. Set the in and out points (by clicking the right facing and left facing arrow buttons) and then click the plus button to add the clip. You can reorder the clips as well. I just wanted you to see this dialog but for this exercise click the Back button so that we don't trim the video. (If you happen to have only a very long video go ahead and make a short clip so the encoding process won't take all day.) Back at the Embedding dialog (first seen in step 3) change the radio button at the bottom to "Embed the entire video." Then click Next.

5. The final screen, Encoding, lets you select the quality and corresponding bandwidth requirements from a variety of preset profiles. Only Flash Professional 8 lets you access the Advanced Settings for the On2 VP6 codec. There you can fine tune the compression settings. Select one of the Flash 7 profiles from the drop down menu and notice the details that appear underneath the menu list Sorenson Spark for the codec. This is the only codec that will work when you publish to Flash player 7. Select "Flash 8 - Low Quality (150kbps)" and you'll be taking advantage of the new On2 VP6 codec—though this also means only users with the Flash Player 8 will be able to view it. The 150kpbs data rate means that the file will use 150 kilo*bits* (not kilobytes) for every second. That is, a 10 second video will be 1500 kilobits. The idea is that if your internet

22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 394

connection can maintain at least 150kbps (for example, most DSL connections are approximately 256kbps) then the video can play without interruption. When you're finally ready to encode click the Next button. You'll see one more dialog reviewing what you're about to and then click Finish.

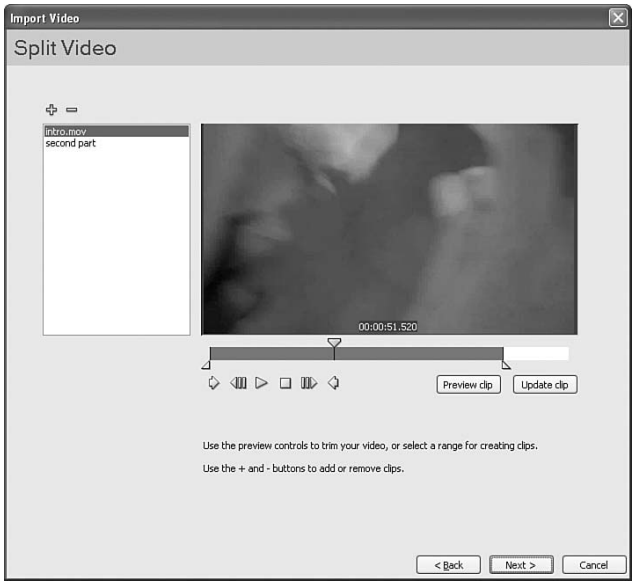

FIGURE 18.3
You can make rough edits to your video during the import process.

6. You'll have to sit through the encoding process (which is, luckily, way longer than the decoding stage at the time the user watches your video). When it's done, the video you just embedded will appear in your Library. Drag the video on to your main Stage. You'll see a dialog box (see Figure 18.4) that asks if you want to extend the Timeline to accommodate the video's total number of frames (in this case, 62). Click Yes.

FIGURE 18.4
When you place a video on the Stage, Flash asks whether you want to extend the Timeline to accommodate the video's length.

22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 395

7. Test the movie so you can hear the audio play. (Note that the sample I used had white on the first frame, so you won't see anything if you're on Frame 1 unless your Stage is a different color.)

If all you wanted was a video inside Flash, you could stop after the preceding task. However, like I mentioned earlier, embedding the video inside your movie the way you did in step 2 above is not always the best choice—creating an external `.flv` is usually better. In addition to the benefits already mentioned, `.flv`s can also be posted on a streaming server (such as Flash Communication Server). This way, users will have hyper access to any point in the video. That is, embedding the video or playing `.flv`s without any special server does what is called progressive download which means users can watch the beginning of a video while the remainder downloads. The premise is that by the time the user needs to view the later portions, those frames will have finally downloaded. In the case of a true streaming server, you can jump to any point in the video and the server sends just those bits. This means users can jump ahead which is not the case in traditional progressive downloading.

I threw in this discussion of the two ways to play `.flv`s (progressive or streaming) because back in step 2 of the preceding task, I promised to explain the other deployment options shown in Figure 18.2. In addition to the Embedded option that you selected in the preceding task, Flash Professional 8 has two "Stream" options plus an option "Progressive download from a web server." (Plus an option for creating QuickTime which is no longer supported in Flash Player 8 so I'm not going to cover it.) Interestingly, the first three options all create the same `.flv`. That is, instead of embedding the video, they output a separate `.flv` file that loads into your movie at runtime. The only difference between these three options is how they pre-populate the parameters for the FLVPlayback component. Basically, to produce an `.flv` the process is nearly identical to the task above—you're just outputting an `.flv` instead of a new item in your library. In addition, you get to select which FLVPlayback skin to use as shown in Figure 18.5. If you have Flash Professional 8 you can do exactly this in the task later this hour "Create an `.flv` and Use the FLVController Component."

22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 396

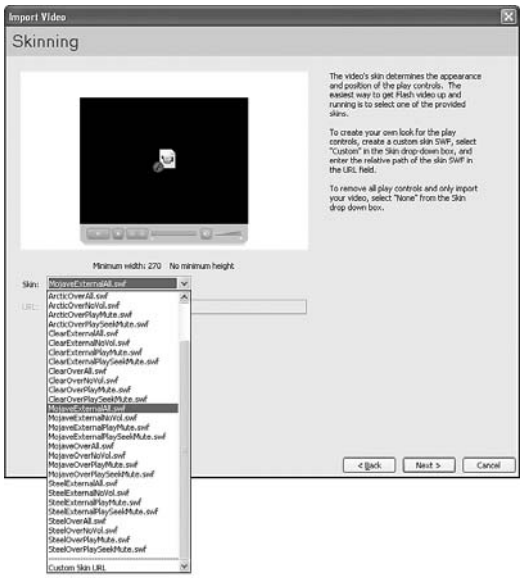

FIGURE 18.5 When using Flash Professional 8 to produce an `.flv` you can select from a variety of skins for the FLVPlayback component.

Using Video

Next you'll learn some of the ways to display videos—both embedded into your `.swf` and created as an `.flv` that resides outside of your file. In this section you'll see how to control the playback of your videos (like giving the user a stop and play button) as well as using special effects such as masking. The creative side of displaying a video is much more interesting than the technical details. But, because the technical details vary between embedded and external video I've separated the two.

Playing an Embedded Video

As a review, a sound in a keyframe that is set to Event begins to play when that frame is reached, and it 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 at this point is that an embedded 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. Flash will tell you exactly how many frames are needed if you drag an embedded video into a timeline that's too short. Another reason to remember Stream sounds is that by default all the audio in an embedded video is affected by the global publish setting for stream audio (which

22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 397

you set by selecting File, Publish Settings to open the Publish Settings dialog box and then selecting the Flash tab).

The visuals in your video behave like Graphic symbols. In fact, embedded 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, as you can other symbols. However, of the symbol types, 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 (as you did in step 4 of the preceding task).

These concepts are just discussed here to help you understand a few of the technical details that follow. You're about to see how easy it is to build controls for an embedded video.

Try It Yourself

Make a Playback Controller for an Embedded 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 as you did in the task "Import a Video" earlier this Hour. That is, make sure you select embed on the deployment dialog that appears when you import and compress the video. Then drag it onto your main timeline and let Flash add frames needed.
2. You're about to create buttons (Stop, Pause, and Play). Put those buttons in their own layer on the main timeline. Lock the layer the video is in and then make a new layer for the buttons.
3. Instead of drawing your own buttons select Window, Common Libraries, Buttons. Inside this Library is a folder called `Playback`. Drag each of the following buttons from that folder to the Stage: Gel Pause, Gel Right, and Gel Stop. You can align these buttons underneath the video any way you want (consider using the Align panel).
4. Open the Actions panel and select the gelPause instance. Then type the following code:

```
on(release){
 stop();
}
```

For a review of ActionScript, revisit Hours 15 an 16.

22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 398

5. Keep the Actions panel open but select the gelRight button instance and type this code:

```
on(release){
 play();
}
```

6. Finally, for the gelStop button, type this code:

```
on(release){
 gotoAndStop(1);
}
```

7. Select the keyframe, and type this code into the Actions panel:

```
stop();
```

8. Test the movie, and you'll be able to control the embedded video.

By the way, it's very easy to add step forward and step backward buttons. Just use this code for your step forward button:

```
on(press){
   nextFrame();
}
```

And this for your step back button:

```
on(press){
   prevFrame();
}
```

One fair criticism of the previous task is that it sure did dirty up the main timeline. That is, the timeline has as many frames as the video does. Normally I'd suggest putting the video inside a Movie Clip; let that clip grow as long as it needs to be; then place the Movie Clip instance in the main timeline where it will only use 1 frame. The problem with that approach is that when Flash reaches the frame containing the Movie Clip it must download all the frames contained in that clip before displaying anything. If the clip has a big video this could mean a long delay. So, doing it the way you did in the task is appropriate for embedded videos. By the way, you can save the preceding task as built and then, in Hour 20, use the MovieClipLoader to load the movie at runtime.

Before we move onto playing external `.flv` videos, I should mention that all the cool stuff—such as masking the video into an odd shape—works with embedded videos. This is covered later this hour in the Special Effects section.

22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 399

Playing an External `.flv` Videos

Earlier this Hour you heard that Flash can play external `.flv` files. Naturally, you need to first create the `.flv`. There are four general ways to produce an `.flv`:

- Import a video into Flash Professional 8 and select one of the first three deployment options listed (that is, not the option to embed the video). The advantage of doing it this way is that it also configures a component for you.
- Use an outside application to produce the `.flv`. Either the Flash 8 Video Encoder (that ships with Flash Professional 8) or the third-party products On2's Flix or Sorenson's Squeeze for Flash.
- Export directly from a supported video editor on a machine with Flash Professional 8 installed. (More about this option and which applications are supported are in the section, "Using Outside Video Editors.")
- Use any version of Flash 8 (including Basic) to first embed the video and then export an `.flv` via the video item in the library. By far, this renders the lowest quality but it works if you don't have Flash Professional 8 and want to follow some of the tasks that would otherwise require the Pro version.

The tasks that follow concentrate on how to play an external `.flv` once it's produced. In the next task, "Create an `.flv` and Use the FLVController Component," you'll use Flash Professional 8 to generate the `.flv` and use it immediately. In "Create and Play an `.flv` the No-Frills Way," you'll use any version of Flash to export an `.flv` and then play it externally without any components. Realize you can use any `.flv` (even if it's not the best quality possible) and then replace it later with a better one.

By the Way

What's Wrong With Videos Produced in Flash Basic 8?

When embedding video inside Flash Basic 8 the minor disadvantage is that you must select a compression profile from a variety of presets—Flash Professional 8 has the advanced settings where you can fine tune things. In every other way, embedded videos are the same for either product. However, embedded videos (even if you export it as an `.flv` from the Library) always use CBR (Constant Bit Rate). A video using VBR (Variable Bit Rate) is always better quality than an identically sized CBR video. That's because every frame in a CBR is the same size. A VBR will use more bandwidth of the frames that need it (say, ones with detail) and make up for the increased size by sacrificing frames that don't need the bandwidth (perhaps blurry or frames with lots of motion). Producing `.flv`s using the first three options above (that is, not embedding and exporting) use VBR.

22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 400

However you produce the `.flv`, playing an external `.flv` will definitely give you the best results—most notably the audio and picture remain synchronized. Playing external `.flv`s is a bit more involved, however. For one thing, you have to remember to upload both the `.swf` and the `.flv` file (plus an additional `.swf` if you use one of the FLVPlayback component's skins). Plus, you only get a preview of the video frames when the video is embedded. If you want to draw animations on top of live action video, you'll need to work with embedded video (at least at the stage where you're producing the animated overlay—you can delete the embedded video once you get the animation done). All I'm saying is that there are additional restrictions when playing `.flv` files.

The following task will probably seem suspiciously simple after the preceding explanation. You'll need Flash Professional 8 for this task, but in it you'll experience the easiest, fastest, and most advanced way to play `.flv`s.

Try It Yourself

Create an `.flv` and Use the FLVController Component

In this task, you'll create an `.flv` and advanced controller with no programming. Here are the steps:

1. Make a new folder in a known location so that all the files you'll need to track are easy to find.
2. Create a new file and save it in the above folder as "my_movie_player.fla." Select File, Import, Import to Stage. Point to a source video file such as a QuickTime.
3. On the Select Video dialog, click Next; on the Deployment dialog, select "Progressive download from a web server" and click Next; on the Encoding dialog, select "Flash 8 - High Quality (700kbps)" (and, if the video is really long, trim the length by dragging the end point triangle next to the preview to the left), then click Next.
4. You should see the Skinning dialog as in Figure 18.5. Select the skin "ArcticExternalPlaySeekMute`.swf`" (or, any one that strikes your fancy with a name ending "...ExternalPlaySeekMute`.swf`"). Click Next. Finally, feel free to read the Finish Video Import dialog and then click Finish.
5. Sit back and wait for the compression to complete. (It can take a long time as the VP6 codec is very asymmetric meaning it takes a long time to compress and much less time to decompress.) When it's done compressing open the folder where you saved your movie and notice a new `.flv` file.

22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 401

6. Back inside your Flash movie you should notice the FLVPlayback component sitting onstage. Select that Component instance and open the Parameters panel. There are lots of parameters you can feel free to modify—including the skin (in case you want to use a different skin than the one selected in step 4). The main parameter, which has already been set for you, is the contentPath (which, you can also change if you change the `.flv`'s file name).
7. Now for the fun part. Test the movie. Pretty sweet how that component works. When you're done watching your video there's one last step that's super important to understand.
8. Go back to your file folder and notice that in addition to the `.flv` and the my_movie_player`.swf` based on your `.fla`, there's another `.swf` (ArcticExternalPlaySeekMute`.swf`) for the component. You need to upload all three (plus an .html file when you publish) when you deploy this to the Web. (More about publishing in Hours 19 and 24, but don't forget that piece for the component—plus the `.flv` file.)

As great as the FLVPLayback component is, you still might want to make your own controls. The FLVPlayback component can be set to no-skin and you can use ActionScript to make your own buttons control the video. While it's actually a very powerful component (even without the skin) you still need Flash Professional 8 to use it. If you only have Flash Basic 8 you can still play external `.flv`s produced elsewhere—just not using the FLVPlayback component.

I should note, too, that the FLVPlayback component requires your users have the Flash 8 Player. This is also a requirement any time you select the On2 VP6 codec. If you want to deliver to the Flash 7 Player not only do you have to select a different codec (Sorenson Spark) but you have to either use one of the Media Components (also only available in Flash Professional) or use no component.

The following task shows how to play an external `.flv` without using components.

Try It Yourself

Create and Play an External `.flv` the No-Frills Way

If you already have an `.flv` produced with Flash Professional 8 or one of the third party tools, you can skip to step 3.

1. Embed a video as you did in the task "Embed a Video." That is, select File, Import and from the second import dialog (the one for Deployment) select Embed Video. If you want this task to work in Flash Player 7, be sure to select

22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 402

a supported profile from the Encoding dialog (everything else you do here won't require Flash Player 8).

2. Once the video is embedded, find the video object in the Library. Double-click on it to access its properties. Click the Export button and save a `.flv` named `file.flv`. You can close this Flash file now because you were just using it temporarily.
3. Start by making sure your `.flv` is named `file.flv`. Create a new movie and save it in the same folder as the `.flv`.
4. Create a video object on the stage. Open the Library window and, from the library's options menu, select New Video. Drag the video object onto the stage and name the instance `my_video`. You can resize the instance to match your actual `.flv`'s dimensions.
5. Create two button instances and name them `stop_btn` and `play_btn`, respectively.
6. Open the Actions panel and click the first keyframe in the timeline, then type this code:

```
my_nc = new NetConnection();
my_nc.connect(null);
my_ns = new NetStream(my_nc);
my_video.attachVideo(my_ns);
play_btn.onPress = function(){
   my_ns.play("my_file.flv");
}
stop_btn.onPress = function(){
   my_ns.play(false);
}
```

There's a ton more you can do with the NetStream object, such as monitor how fast a video is downloading. The only catch is that, unlike using the FLVPlayback component (which also has much more to it), you'll have to do most of the coding by hand. You're welcome to read all about in a Flash Video paper I wrote at www.phillipkerman.com/wholestory.

Special Effects

Now for the fun part. Once you've embedded a video or set up a component of video object to display an external `.flv`, you can perform countless special effects that can dramatically change the way a user experiences your video. We'll look at just a few. Namely, the old technique called rotoscoping where you effectively draw on every frame; matting where you make the video appear inside an odd shape

22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 403

(instead of a rectangle); and including an alpha channel with your video so you can change the background any time (including at runtime). At the end of this section I'll include some code that lets a user select which video they want to watch.

Try It Yourself

Rotoscope (Draw Frames of a Video)

In this task you'll combine frame-by-frame animation with live action video. Here are the steps

1. Embed a video like you did in the task "Embed a Video" earlier this hour. At the end, I'll show you how to convert this task to work with an external `.flv`, but you have to start by embedding the video. If you are planning on playing an external `.flv` select Modify, Document and set the framerate to match the framerate at which you're going to render the video.
2. If you're planning on leaving this as an embedded video, place the video object in the main timeline. If you're planning on the converting this to an `.flv` then first make a new Movie Clip and put the video object inside the clip. In either case, make sure you say "OK" to the dialog asking to add more frames to accommodate the video's duration (as earlier in Figure 18.3).
3. In the timeline where the video appears, insert a new layer (which should appear above the video's layer). Ensure the new layer is both above the video and extends the entire length of the video. (If not, just move the layer and click a cell above the last frame in the video and press F5.)
4. Click the layer name of the empty layer to select the entire span of frames and press F6. You'll now have an empty keyframe above each frame of the video where you can draw.
5. Select the Brush tool and pick a bright color. Hold the mouse in one hand and put the other hand on the > button. Draw right into the empty keyframe of frame one (perhaps draw an outline around an object in your video such as a person's face). After you draw press > and draw another frame. Even if you have hundreds of frames to draw you can do it quickly. It's definitely possible to insert pauses (by removing the keyframes) or even do tweens that match the video, but it might be just as fast to simply draw every frame. For this exercise feel free to stop after 10–20 frames and come back after you've had some practice to finish them all.
6. If you're going to leave the video embedded, you're done. Go ahead and test the movie. If you want this animation to work with an external `.flv` you've

22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 404

got two steps left. First, remove the video from the timeline and create an `.flv` named `my_file.flv` (as you did in "Create an `.flv` and Use the FLVController Component" if you have Flash Professional 8 or as you did in "Create and Play an External `.flv` the No-Frills Way" if you only have Flash Basic). And second, write the code so that the animation syncs up with the external `.flv`.

7. Instead of actually deleting the video, simply access its layer properties and set it to Guide so that you'll always have the video for reference. If the video object isn't being used anywhere then it won't add to the file size. Drag on to the stage an instance of the Movie clip containing your animation. Name this instance `my_animation`.

8. From the options menu in the Library select New Video. Drag an instance onstage and name the instance `my_video`. Select the first keyframe in the movie, open the Actions panel and type this code:

```
my_nc = new NetConnection();
my_nc.connect(null);
my_ns = new NetStream(my_nc);
my_video.attachVideo(my_ns);

my_animation.stop();
my_ns.play("my_file.flv");

var framerate=12;
onEnterFrame=function(){
   my_animation.gotoAndStop(Math.floor(my_ns.time*framerate));
}
```

Note that you'll need to set the `framerate` variable to match your movie's actual framerate (shown here as `12`). The code that runs every enter frame ensures the `my_animation` clip remains in sync with the time of the NetStream (that is, the external video).

There's a lot more you can do with synchronizing animation with a video. In the task above, every frame of the animation was synchronized with a frame in the video. Often, you only need to update an overlaying graphic once in a while. For example, you could display a caption containing the speaker's name or bullet points that match what the speaker is discussing. In such situations you don't need *every* frame synchronized. Such cases are more appropriate for cue points. In Flash Professional 8 you can actually inject cue points into an `.flv` at the time you encode it (Figure 18.6) or use the Media Components to set cue points (shown when you do the next task in Figure 18.7).

22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 405

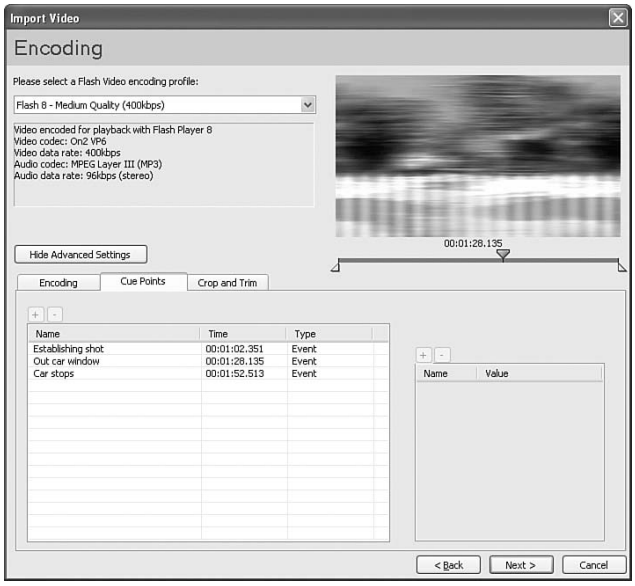

FIGURE 18.6
You can inject cue points right into the `.flv` when you encode using Flash Professional 8.

Inserting cue points is fairly intuitive. In fact, there's a third-party tool called Captionate (www.captionate.com) which I'd highly recommend for injecting `.flv`s with cue points and complete captioning data. In all cases, the code to actually make something happen when the cue points are triggered is slightly more involved. In the following task you'll set up a Movie Clip to display bullet points in synch with a video. After you have it set up, you can take one of three approaches: synch to cue points injected into the `.flv`, manually input cue points into the MediaPlayback component, or do everything manually (the old-school way).

Try It Yourself

Use Cue Points

You can download an `.flv` for this exercise from www.phillipkerman.com/teachyourself/samplefiles/.

1. Create a new Flash file and save it adjacent to the `colors.flv` you downloaded.
2. Select Insert, New Symbol and name it Bullets and pick Movie Clip then press OK.

22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 406

3. You should be inside the Bullets Movie Clip. Create some static text that reads "Blue." Click the keyframe and use the Properties panel to type `blue` into the frame label field. Insert a blank keyframe in frame 2.

4. In frame 2 we'll do something a bit more fancy. Namely, we'll place a symbol which itself contains an animation that plays once and stops. Start by creating some static text that reads "Pink Stuff." Select the text and choose Modify, Convert to Symbol. Name it "Pink Text" and make sure it's a Movie Clip. Now, with that instance of Pink Text selected, select Modify, Convert to Symbol (again). Name it "Animated Text" and make sure it's a Movie Clip. Now, double-click the instance of Animated Text. Inside this clip click on frame 10 and insert a keyframe (press F6). While frame 10 is selected, open the Actions panel and type the code: `stop();` Go back to frame 1 and move the instance to the left and set the alpha to 10%. Click the keyframe in frame 1 and from the Properties panel select Motion. Go back up one level (so you're inside the Bullets symbol). Finally, give frame 2 (of the Bullets symbol) the label `pink`. Now when the user arrives in frame 2 they'll see the "Pink Stuff" text animate into place and stop.

5. Inside the Bullets symbol, go to frame 3 and insert a blank keyframe (press F6). Type some static text that says "Green Leaves". You're welcome to make this animate too—just make sure you nest any animation. When you're done be sure to give the third frame (here inside the Bullets symbol) the label `green`.

6. Place an instance of the Bullets symbol on stage and give it an instance name `bullets`.

7. Here's the cool part. You can take any one of the following three steps: You can follow step 8 that uses the new FLVPlayback to sync the bullets to the cue points that I already embedded into the `.flv`. Alternatively, go straight to step 9 to use the older Media Components to input the sync points manually (which will work with any `.flv`). Finally, if you don't have Flash Professional 8, you can jump to step 10 where you can use a video object and use an array—definitely the most home rolled approach (but it works with Flash Basic 8).

8. Approach 1: Drag an FLVPLayback component on to the stage and place it next to the `bullets` instance. Give the FLVPlayback an instance name `playback`. Select the first keyframe and type this code into the Actions panel:

```
playback.contentPath = "colors.flv";
bullets.stop();
myListener = new Object();
myListener.cuePoint = function(evt) {
  bullets.gotoAndStop(evt.info.name);
};
playback.addEventListener("cuePoint",myListener);
```

22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 407

9. Approach 2: Drag a MediaPlayback component onto the stage and place it next to the `bullets` instance. Give the MediaPlayback an instance name `playback`. Select Window, Components Inspector and select the MediaPlayback instance. Click the plus button three times to add three sync points. Name the three sync points blue, pink, and green respectively. Set the times to `0:0:0:0`, `0:0:4:4`, `0:0:7:2` respectively. The populated component inspector should look like Figure 18.7. Finally, select a keyframe in your movie and type this code into the Actions panel:

```
playback.contentPath = "colors.flv";
bullets.stop();
myListener = new Object();
myListener.cuePoint = function(evt) {
  bullets.gotoAndStop(evt.cuePointName);
};
playback.addEventListener("cuePoint",myListener);
```

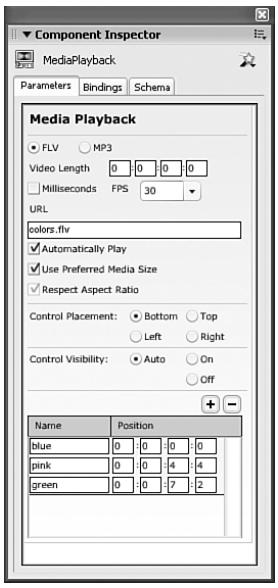

FIGURE 18.7
The MediaPlayback lets you set cue points from inside Flash.

10. Approach 3: From the Library's options menu, select New Video. Drag an instance of the video object on to the stage. Give it an instance name `my_video`. Open the Actions panel, select a keyframe and type this code:

```
bullets.stop();
my_nc = new NetConnection();
my_nc.connect(null);
my_ns = new NetStream(my_nc);
my_video.attachVideo(my_ns);
```

22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 408


```
myPoints=[0, 4.4, 7.2];
myLabels=["blue", "pink", "green"];
onEnterFrame=function(){
  if ( my_ns.time > myPoints[myCounter] ){
    bullets.gotoAndStop(myLabels[myCounter]);
    myCounter++;
  }
}
//every time you want to restart the video, do these three lines:
myCounter=0;
bullets.gotoAndStop(1);
my_ns.play("colors.flv");
```

If you don't mind requiring Flash Player 8, the first approach (step 8) is best. If you have Flash Professional and don't use the On2 codec you can deliver to Flash Player 7 using the second approach (step 9). Finally, with Flash Basic 8 you're limited to the last approach (step 10). In that case, you can decide whether the On2 codec warrants requiring Flash Player 8.

Optimizing Quality and File Size

With all this talk of encoding options and coding tricks it might be easy to lose sight of the core goal: namely, to produce the best looking video that downloads fast. It may seem quaint to study traditional techniques (developed decades ago) but that's exactly what you should do if you want good looking video. For example, use a tripod and shade the camera lens to reduce flare which can desaturate the colors. Consider a few strategic cuts instead of special effects that can make transitions long and arduous. The point is there's a wealth of experience photographers and film makers can share that all translate to digital video.

I also have a few technical tricks that can reduce the file size without having a huge impact on quality. The two biggest factors that have an immediate impact on file size are the video's framerate and its pixel dimensions. For example, a 12 fps video will be nearly exactly half the size as a 24 fps video. A lower framerate will not only be smaller, but it won't look quite as good—especially if there's a lot of motion. The best thing to do is to take a small representative sample and test different framerates. Keep going lower and lower until the quality is unacceptable then back off.

Just like any raster graphic, you can also render videos at different dimensions. Similar to how low framerates make for a smaller video, smaller dimensions make the file smaller too. But here's a great tip that can have a surprising effect. Often, you can render a video at half its final size and then stretch it during playback. For example, you want a video to display at 240×320. You can render it at 120×160 but just stretch the video holder (video object or component) to 240×320. Test it out! Make a video at 240×320 at a particular bandwidth and make another at the same

22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 409

bandwidth but only 120×160. Naturally, the smaller one will be sharper until you stretch it. But the paradox is that often the stretched one looks way better than the same file sized unstretched one.

Lastly, as a bit of a repeat from Hour 10, "Including Sound in Animations," stereo sound is twice as big as mono. Just be sure you really need stereo before you include it in your video. By the way, when you embed video inside your Flash movie, you set the compression level via the Publish Settings for Stream sound.

Using Outside Video Editors

Flash Professional 8 comes with both the Flash 8 Video Exporter and a plugin to let other video editors create `.flv`s directly. These get installed automatically when you install Flash (or you can run the installer later) and it works in conjunction with various popular video editors. The idea is that video professionals can best make final edits and other sweetening in their favorite video editor and then export directly to the Flash video format (`.flv`). In addition, you can use the stand-alone version of the Flash 8 Video Exporter to compress raw videos into `.flv`s in batches.

To use the plugin you just need to launch one of the supported video editors on the same machine where you have Flash Professional 8 installed. The supported software includes Adobe After Effects, Apple FinalCut Pro, Avid Xpress DV, and Discreet Cleaner among others. Once you're finished editing the video, simply select something like File, Export, Flash Video (though the exact menus differ for each product). You'll see a dialog identical to the Encoding dialog you saw when you used Flash to do the compression.

The stand alone version should be installed in a folder adjacent to where Flash 8 Professional is installed (C:\Program Files\Macromedia\ for example). Again, the available options are identical to those found when using Flash to perform the compression. However the stand alone version adds a batch feature that will, at your convenience, compress a long list of videos you've added to the queue. This means you can take several videos, add them to the queue—even add the same one but select different compression settings for comparison—and then let them compress overnight. Video compressors are always slow and the On2 VP6 codec is extra slow when compressing.

Summary

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. The majority of this hour dealt with the technical limits of video. Don't

22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 410

let that stop you. The price (extra work on your part and extra download time for the user) can definitely be worth it when necessary.

In this Hour you learned how to embed a video right into your file as well as how to create a Flash video file (`.flv`) and play it externally. Embedding a video may be easier and does provide the ability to step through each frame but when the video is longer, the added file size (to your main movie) isn't worth the cost of embedding it. Keeping an external `.flv` not only provides the best synchronization and video quality but the user only downloads the video's they request. There really are many more topics worth studying in Flash Video but this chapter touched on the primary points.

Q&A

Q ***I realize that you recommend against using*** `.avi` ***files, but that's all I have. Should I first convert an*** `.avi` ***file to a QuickTime file by using a tool such as the QuickTime player?***

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

Q ***I've embedded a video with a great music track. Why does the audio sound so terrible?***

A You need to set how the audio is to export via the Stream compression in the Flash tab of the Publish Settings dialog box. See Chapter 10 for more about sound compression. The key here is that the Stream setting is what affects the audio in an embedded video.

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

A 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 on the Flash tab) 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.

22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 411

Q ***Every time I attempt to compress a particular video (regardless of whether I select to make an external*** `.flv` ***or embed the video), Flash reports that the audio can't be imported. What's the problem?***

A Depending on the type of audio track in your video, Flash may simply not support it. You'll need to get an new source file.

Workshop

The Workshop consists of quiz questions and answers to help you solidify your understanding of the material covered in this hour. You should try to answer the questions before checking the answers.

Quiz

1. If On2's VP6 coded yields better results than the Sorenson's Spark codec (which it does), why would you ever select Spark?

   A. You don't want to pay the additional license charges which accompany the VP6 codec.

   B. You are planning on delivering to Flash Player 7 and the VP6 codec only works on Flash Player 8 or later.

   C. You don't like the fact the VP6 encoded videos are at least twice the size of Spark encoded videos.

2. How do you change the compression on a video that you've already embedded?

   A. You can't. Instead, you could re-embed and recompress at that time.

   B. You simply access the video item (in the Library), select its Properties option, and then click Recompress.

   C. You simply need to modify the Video tab of the Publish Settings dialog box.

3. On2's VP6 is which of the following?

   A. An old technique where rock bands would begin to play "on two" instead of "on four" as in "one, two, three, four"

   B. The compression technology included with Flash 8 and the decompression technology used in the Flash player

   C. An option (that costs extra) to compress your videos with "supercompression"

22_0672327546 CH18.qxd 9/7/05 7:39 AM Page 412

Quiz Answers

1. B. Requiring users to upgrade to the latest player is definitely an issue, especially when it's so new.
2. A. Answer B makes sense, in fact most imported media lets you reimport via the Update button on the item's Properties dialog box—but that is not supported for video and simply displays a warning.
3. B. A codec has 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.