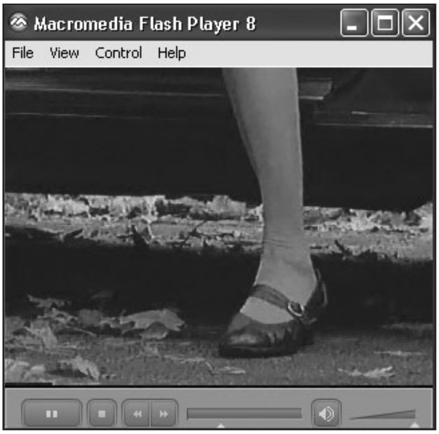
7 Creating Flash video

In 2004, Macromedia started popping up at conferences showing off some of the really interesting things one could do with video using the Flash Player. Prior to that, developers were quietly discovering that Flash Player 7 was a serious media player that required no extra software to play. About that same time, Jeremy Allaire, one of the inventors of ColdFusion, told anybody who would listen that Flash Player was a Trojan Horse and “people are just now turning it on.”

What makes this technology so popular is the ubiquity of Flash Player. None of the video player solutions–from QuickTime to Real–have the market penetration nor have experienced the rapid adoption of Flash Player. Flash Professional 8 builds upon the success of the past and puts video creation and delivery squarely into your hands.

This lesson will begin with the use of the new video encoder tool and gives you the knowledge necessary to professionally produce video that meets the needs of the users. It will end with your creating a Flash video using the new video PlayBack component.



The video and the controls used in this lesson.

What You Will Learn

In this lesson you will:

- Use the Video Wizard to create a FLV file
- Choose a skin for a Flash 8 video component
- Use the Advanced Encoding settings in the Flash 8 video encoder
- Customize the data and audio rates and set the keyframes in a Flash video
- Learn the difference between streaming video and a progressive download
- Attach a FLV file to the FLVPlayback-Player 8 component
- Use the Component inspector to set a video component's parameters
- Buffer video for optimized Web playback
- Attach video to a Flash 6-7 MediaPlayback component
- Prepare a Flash video for playback through Flash Player 7
- Test a video
- Use the FLV UI components to control video playback

Approximate Time

This lesson will require about one hour to complete.

Lesson Files

Media Files:

MAMMAAND.MOV

Starting Files:

None

Completed Files:

Lesson07/Complete/Wizard.fla
Lesson07/Complete/Component.fla
Lesson07/Complete/ComponentV7.fla

Using Video in Flash

The inclusion of video into your Flash projects will result in experiences that are compelling and engaging. You can add video to a Flash movie or simply create a Flash movie containing nothing more than the video for inclusion in a Web page. However, keep in mind that creating video involves more than deciding to add a video to a Web page or Flash movie.

The first thing to keep in mind is the user. Video is *rich media*—it enriches the user experience…if it is done right. Though Flash might be a cool technology and video—even cooler, if a user has to wait for an inordinate amount of time for the video to load—the rich experience will turn sour in a very short order. Never forget that your video is being delivered through the Internet, and a key factor is the user's bandwidth. If you create a FLV file for broadband playback, you can bet that users with a dial-up connection are in for a long wait!

Apart from keeping an eye on the pipe, you should also pay close attention to the user, who might not be interested in the video—so the inclusion of an on/off control is an important consideration. Give users control of the entire experience by letting them make the decision whether or not to play the video by not having the video play when it loads. If it is a large video, include a progress bar that shows the user that the video is indeed loading. Controls such as a volume control and a scrub bar that advances the video as you drag it are other techniques you can use to let the user control the process.

Using the Flash Video Encoder Wizard

The first thing you have to understand about the use of video in Flash is the original video must be converted to a format Flash can read. This format is FLV (Flash Video) and you can either use Flash, the Flash Video Encoder, or third-party software to create the FLV file. If you use video-editing software such as Adobe After Effects, you can also export the file as a FLV file. If you are a Dreamweaver user, the current version of Dreamweaver—Dreamweaver 8—contains a feature that allows you to add an FLV at the click of a button. The purpose of the Encoder, therefore, is to convert the video file to the FLV format.

When you installed Flash Professional 8, the Flash Video Encoder application was also installed on your machine. If you also have QuickTime, Adobe After Effects, Avid Studio, or Final Cut Pro installed on your machine, they, too, were given the capability to use the Encoder when it was installed.

If you will be converting a video to the FLV format, the video must be in one of these formats:

AVI—Audio Video Interleave

MOV—QuickTime 4 or later

MPEG—Motion Picture Experts Group

DV—Digital video, the format used by most commercial camcorders

WMF/ASF—Windows Media

Note *Don't be terribly surprised if, in certain instances, Flash imports the video but not the audio. For example, audio is not supported in MPG/MPEG files imported with QuickTime 4. In such cases, Flash displays a warning indicating that the audio portion of the file cannot be imported or the audio section will be dimmed in the Encoder.*

For this exercise, you will use Flash's Video Wizard to import a video and format a video that will be placed in a Web page.

1. Copy the Lesson 7 files to your hard drive or create a folder on your hard drive and copy *MOMMAAND.MOV* to this folder.

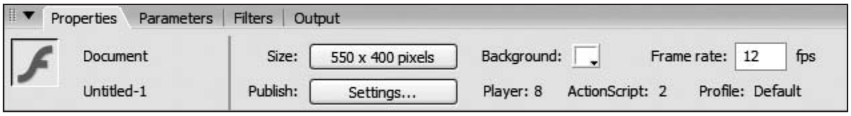
The wizard will be creating a FLV file for you and saving it to the same folder as the source video. If you try to work from the CD, you will get an error.

2. Create a new file called *video.fla* and save it anywhere on your hard drive.

When you create this file use the default dimensions for the Stage, which are 550 by 400.

3. Change the frame rate of your Flash movie to 15 fps.

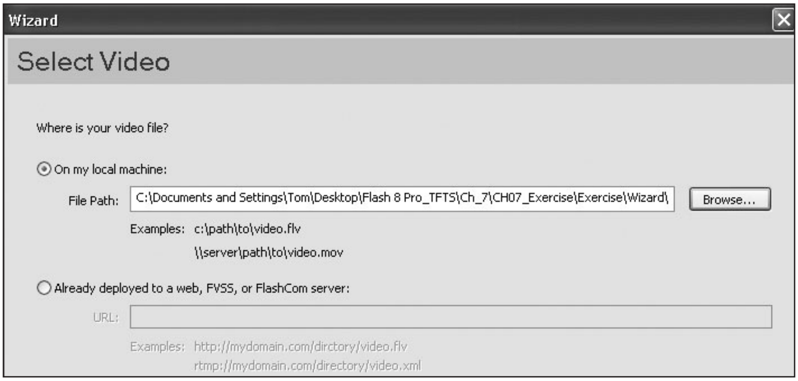
When creating video, a good habit to develop is to have the frame rate of the video and the Flash movie in which it will be placed match. Thus, you avoid having the video finish before the sound and other nasty issues that tend to crop up when there is no match. The frame rate can be changed in the Property inspector.



Document
Untitled-1
Size: 550 x 400 pixels
Publish: Settings...
Background:
Frame rate: 12 fps
Player: 8
ActionScript: 2
Profile: Default

4. Choose File › Import › Import Video to open the Video Import Wizard.

From here, you can navigate to `MOMMAAND.MOV` on your hard drive. The Video Import Wizard leads you through the import and conversion process. When the Select Video page of the wizard opens, click the Browse button and navigate to the folder containing the video. Select the file and click the Open button. The path appears, and you can click the Next button to open the Deployment page.



Select Video

Where is your video file?

● On my local machine:

File Path: C:\Documents and Settings\Tom\Desktop\Flash 8 Pro_TFTS\Ch_7\CH07_Exercise\Exercise\Wizard\

Examples: c:\path\to\video.flv
\\server\path\to\video.mov

○ Already deployed to a web, FVSS, or FlashCom server:

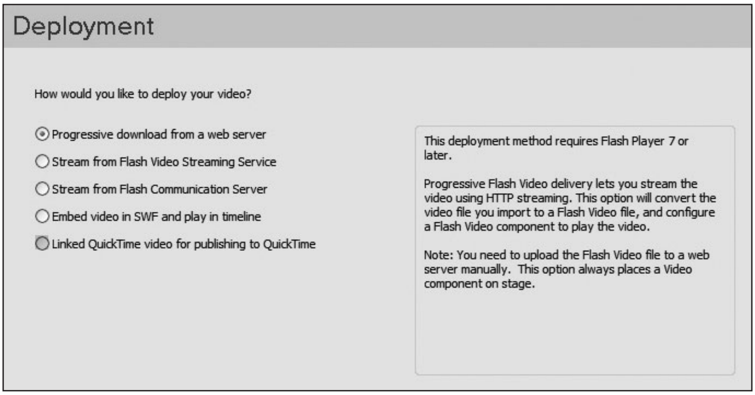
URL:

Examples: http://mydomain.com/dirctory/video.flv
rtmp://mydomain.com/directory/video.xml

Note *The other option–Already deployed to a Web, FVS,S or Flash Comm server–assumes that the FLV file has already been created and is sitting on your Web server, a Flash Video Streaming Service server, or a Flash Communication Server.*

5. Select Progressive download from a standard Web server and click Next.

A progressive download is a standard for Flash video being played from a Web server. In very simple terms, this means that as soon as enough of the video has downloaded into the SWF file on the Web page, the video plays. Depending on the length and quality of the video, this delay ranges from almost instantly to a few seconds.



Deployment

How would you like to deploy your video?

● Progressive download from a web server
○ Stream from Flash Video Streaming Service
○ Stream from Flash Communication Server
○ Embed video in SWF and play in timeline
○ Linked QuickTime video for publishing to QuickTime

This deployment method requires Flash Player 7 or later.

Progressive Flash Video delivery lets you stream the video using HTTP streaming. This option will convert the video file you import to a Flash Video file, and configure a Flash Video component to play the video.

Note: You need to upload the Flash Video file to a web server manually. This option always places a Video component on stage.

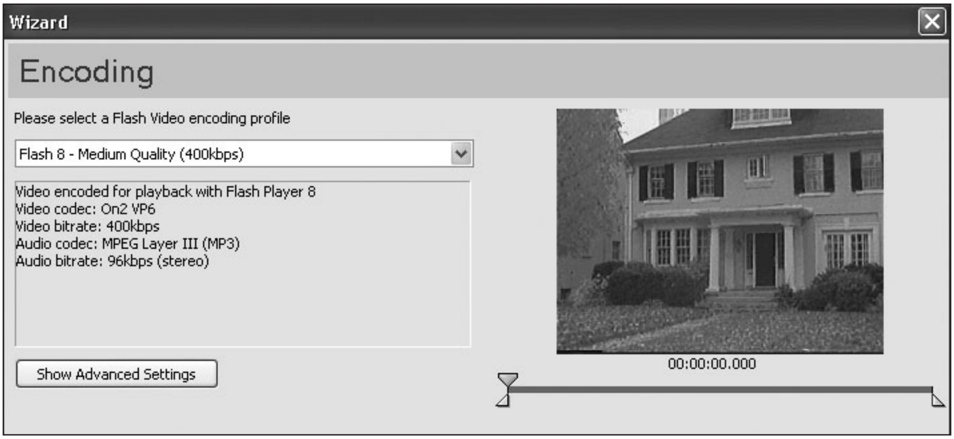
Note *Only under the most extreme conditions should you even consider embedding a video into the SWF file and playing it from the Timeline. If the video is less than 15 seconds this practice is acceptable because the SWF file won't play until all the data has been loaded. This means the user is in for a long wait.*

6. When the Encoding page appears, select Flash 8 - Medium Quality (400kbps) from the encoding profile drop-down list and click Next to move to the Skinning page.

The list of quality options in the encoding profiles allows you to choose both the version of Flash Player to be used and a bit rate. Bit rate determines how fast the video data streams into the SWF file embedded in a Web page. The one chosen here is appropriate for broadband delivery. Low quality is best for dial-up situations, and high quality is ideal for high-speed and LAN connections.

Tip *The three sliders under the video preview can be used to trim the video. The top slider simply shows the position of the playhead in the video. Move it back and forth to scrub through the video. The two sliders under it can be used to set the in and out points for the video. For example, assume that the video has a couple of seconds of black before it starts. Drag the in point to the frame; when the video is encoded, that first couple of seconds of black screen will not appear. The slider on the right is the out point. Drag it to the left, and any video beyond this point will be ignored.*

Note *We'll deal with the advanced settings later on in this lesson.*



Wizard

Encoding

Please select a Flash Video encoding profile

Flash 8 - Medium Quality (400kbps)

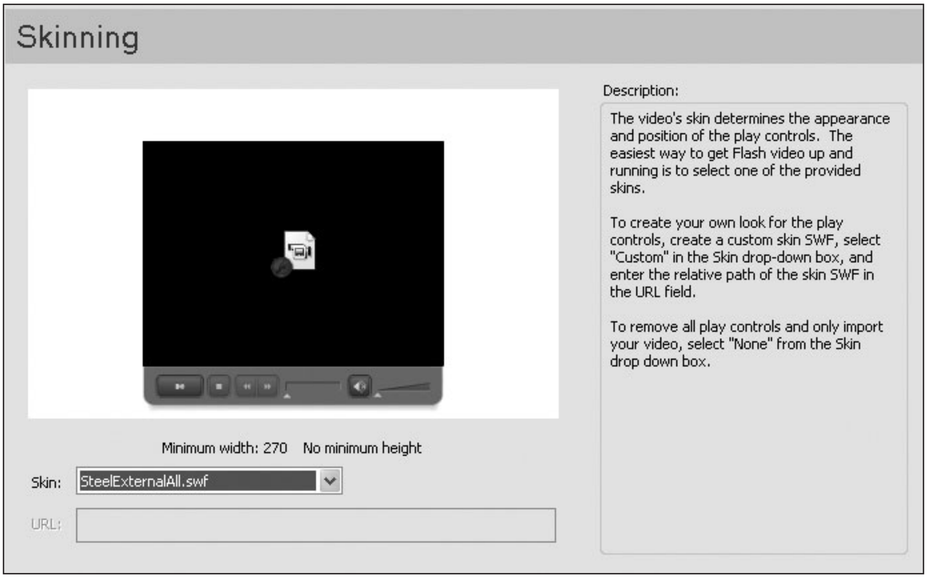
Video encoded for playback with Flash Player 8
Video codec: On2 VP6
Video bitrate: 400kbps
Audio codec: MPEG Layer III (MP3)
Audio bitrate: 96kbps (stereo)

7. In the Skinning page, select *`SteelExternalAll.swf`* from the choices in the Skin drop-down list and click Next to move onto the Finish Video Import page.

The skins, for all intents and purposes, are nothing more than a collection of 32 custom controller styles for the video. The one chosen places the controls under the video and makes them visible to the user. Pay close attention to their names. Each of the styles allows you to place the controller over the video—SteelOverAll.swf. In this case, the video appears on the Web page, and the controls appear only when the user rolls the mouse over the video. Other styles, such as `SteelOverNoVol.swf`, remove controls such as a volume slider from the controller. When you choose a skin, a preview of your selection appears in the page.

Note *The Custom style is ideal for those situations in which you want to create your own controller. For example, you might want to add your client's logo to the controller and use a different style from those offered. This can be done, but the technique is out of the scope of this book.*

Tip *Note the minimum width in the Skinning page, which tells you the minimum Stage width required for the controller. This value can be set in the Property inspector. If the video is wider than the minimum value, you can ignore this direction.*



Skinning

Description:

The video's skin determines the appearance and position of the play controls. The easiest way to get Flash video up and running is to select one of the provided skins.

To create your own look for the play controls, create a custom skin SWF, select "Custom" in the Skin drop-down box, and enter the relative path of the skin SWF in the URL field.

To remove all play controls and only import your video, select "None" from the Skin drop down box.

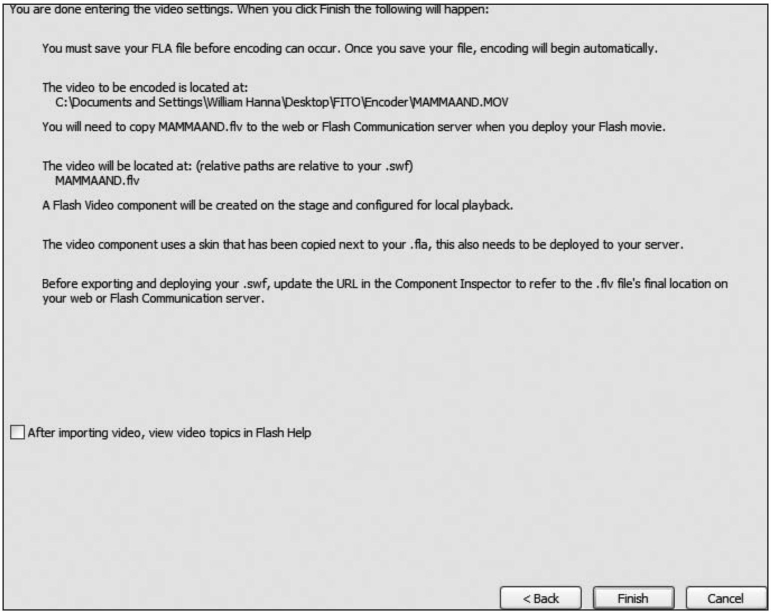
Minimum width: 270 No minimum height

Skin: SteelExternalAll.swf

URL:

8. Examine the Finish Video Import page. When you're sure that all the information on this page is correct, click Finish.

The Finish Video Import page of the wizard gives you all the information you need regarding what happens next. It also tells you where the files it creates will be placed, how to link the video to the SWF file, and even allows you to access the Help file when the import is finished. If you make a mistake or need to change a setting, click the Back button.



You are done entering the video settings. When you click Finish the following will happen:

You must save your FLA file before encoding can occur. Once you save your file, encoding will begin automatically.

The video to be encoded is located at:
C:\Documents and Settings\William Hanna\Desktop\FITO\Encoder\MAMMAAND.MOV

You will need to copy MAMMAAND.flv to the web or Flash Communication server when you deploy your Flash movie.

The video will be located at: (relative paths are relative to your .swf)
MAMMAAND.flv

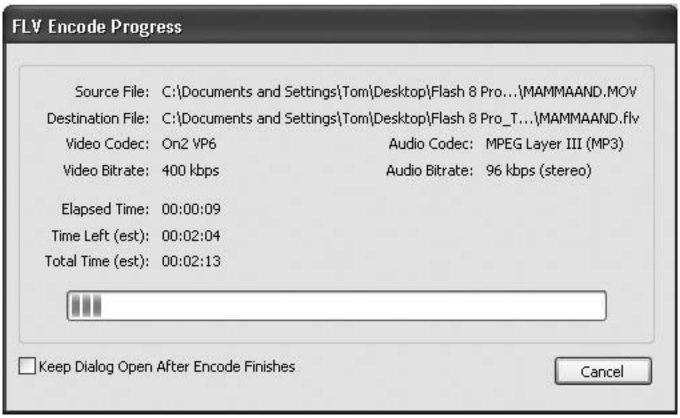
A Flash Video component will be created on the stage and configured for local playback.

The video component uses a skin that has been copied next to your .fla, this also needs to be deployed to your server.

Before exporting and deploying your .swf, update the URL in the Component Inspector to refer to the .flv file's final location on your web or Flash Communication server.

□ After importing video, view video topics in Flash Help

When you click the Finish button, the FLV Encode Progress page appears, showing you the progress of the encoding process.



FLV Encode Progress

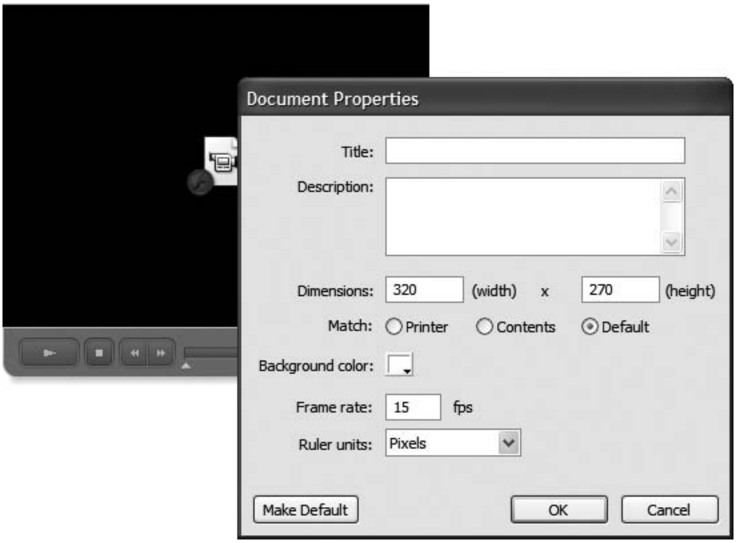
```
     Source File:  C:\Documents and Settings\Tom\Desktop\Flash 8 Pro...\MAMMAAND.MOV
Destination File:  C:\Documents and Settings\Tom\Desktop\Flash 8 Pro_T...\MAMMAAND.flv
     Video Codec:  On2 VP6                    Audio Codec:  MPEG Layer III (MP3)
   Video Bitrate:  400 kbps                 Audio Bitrate:  96 kbps (stereo)

    Elapsed Time:  00:00:09
 Time Left (est):  00:02:04
Total Time (est):  00:02:13
```

□ Keep Dialog Open After Encode Finishes

9. Click the Stage; in the Property inspector, set the Stage dimension to 320 x 270.

If a video is destined for a Web page, always set the Stage size to match that of the video. The unused portion of the Stage is nothing more than excess bandwidth. This video is 320 × 240, but the extra 30 pixels will accommodate the height of the controller under the video.



Document Properties

Title:

Description:

Dimensions: 320 (width) x 270 (height)

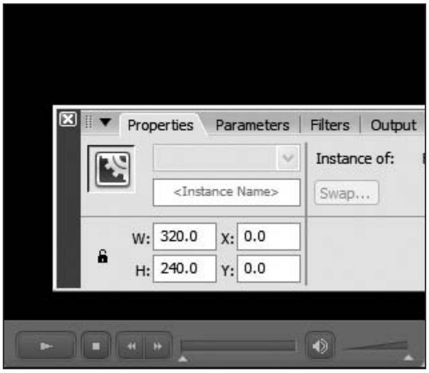
Match: ○ Printer ○ Contents ● Default

Background color:

Frame rate: 15 fps

Ruler units: Pixels

10. Click the video and set the *x* and *y* coordinates to 0,0.

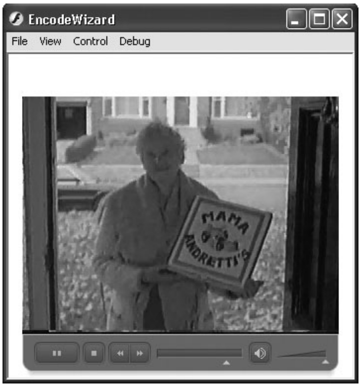


Instance of:

W: 320.0 X: 0.0

H: 240.0 Y: 0.0

11. **Save the movie and test it.**

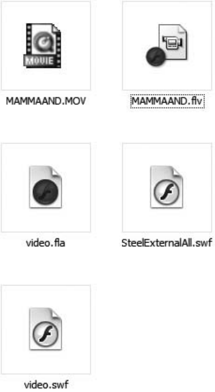


12. **Publish the movie. Minimize Flash and open the folder where you have saved this Flash project.**

Inside the folder are a number of files:

- `video.swf`
- `SteelExternalAll.swf`
- `MAMMAND.FLV`
- `Video.fla`
- `MAMMAND.MOV`

The first three files—`video.swf`, `SteelExternalAll.swf` and `MAMMAND.FLV`—are the ones that have to be uploaded to the Web server. The one that gets placed in the Web page is `Video.swf`.



Using the Flash 8 Video Encoder and the Advanced Settings

Although the wizard is an extremely useful tool, especially for rapid prototyping, the true power of Flash video is realized by the capability to create a FLV file that is optimized for the user's bandwidth. The video you created in the previous exercise, for example, is ideal for playback through a cable modem, but will give the users of dial-up services a less-than-satisfactory experience.

The Flash 8 Video Encoder asks you to answer some very specific questions as you move through the process of encoding the video. Among them are these:

- Which codec do you want to use?
- At what data rate do you want the video to stream?
- What is the data rate for the Audio?
- Is the video to be resized?
- How often will keyframes be added?

The encoder, like the wizard, creates a stand-alone FLV file that can be uploaded to a variety of servers.

In this exercise, you create the same video player as that from the previous exercise, but there are a couple of differences. You manually add the FLVPlayback-Player 8 component and connect the FLV file to it in Flash. You also custom-optimize the FLV file for broadband playback rather than use a preset.

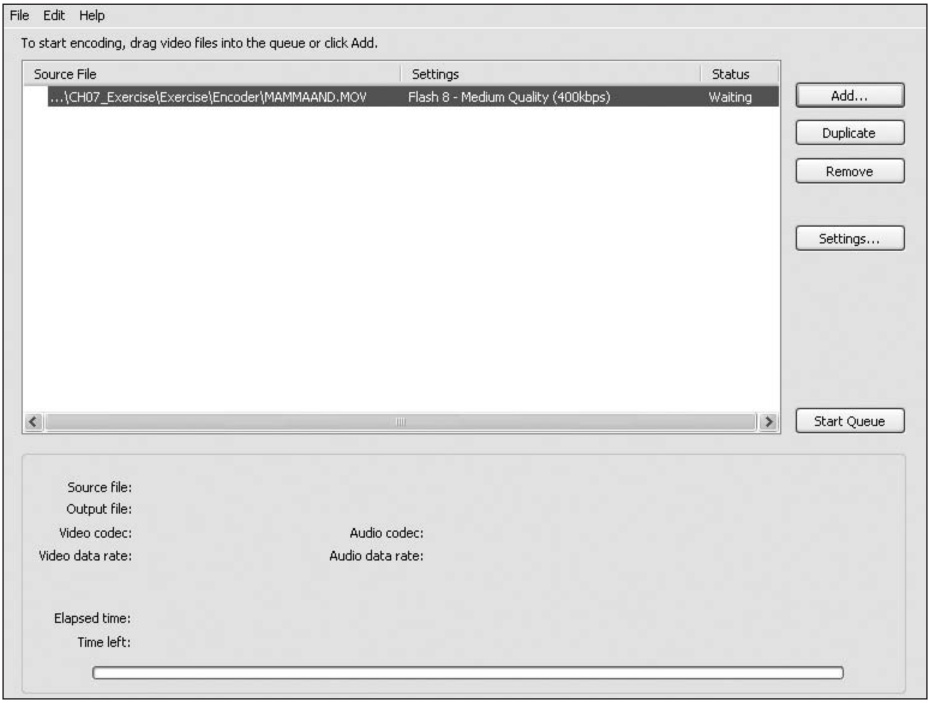
Before you start, be sure that you have copied the Lesson 7 folder from your CD to your hard drive.

1. Open the Flash 8 Video Encoder.

If you are on a PC, you can access this application by opening the application located at `C:\Program Files\Macromedia\Flash 8 Video Encoder`. Mac users can find the application at `Macintosh HD\ Applications\ Macromedia\ Macr Flash 8 Video Encoder`.

2. Drag the *`MOMMAND.MOV`* file from your Lesson Folder to the files area of the Encoder. Alternatively, you can click the Add button and navigate to the folder and select the video.

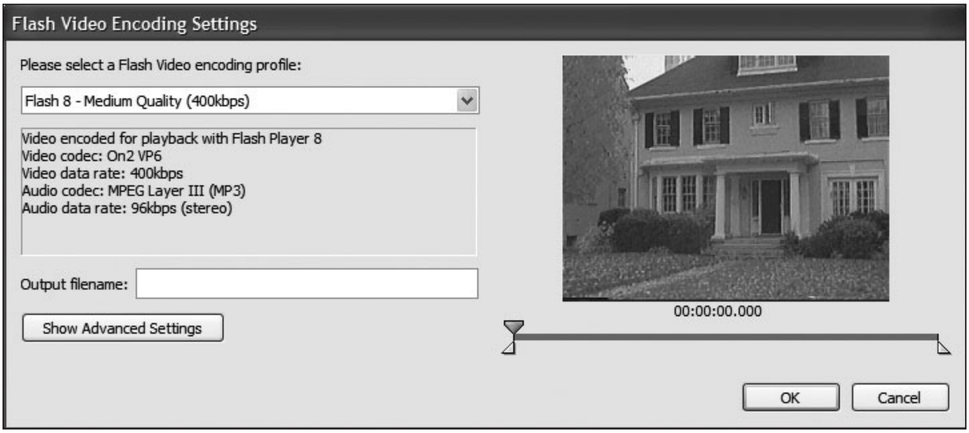
When you add a video, the path to the file is shown in the Files area. As well, the default encoding settings are applied to the video file.



To start encoding, drag video files into the queue or click Add.

Source File	Settings	Status
...\CH07_Exercise\Exercise\Encoder\MAMMAAND.MOV	Flash 8 - Medium Quality (400kbps)	Waiting

3. **Click the Settings button to open the Encoding Settings window.**

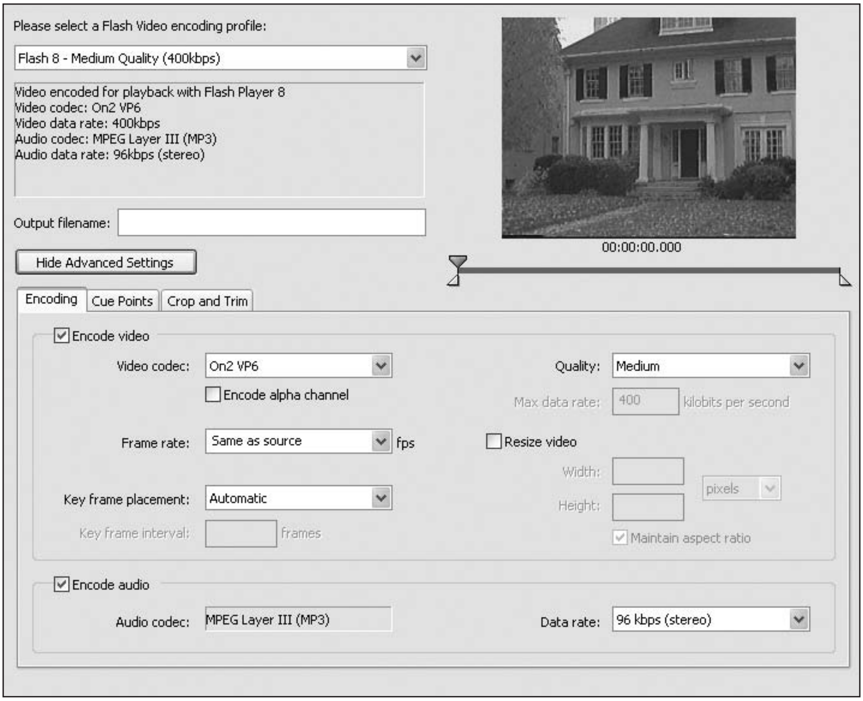


Please select a Flash Video encoding profile: Flash 8 - Medium Quality (400kbps)

Video encoded for playback with Flash Player 8
Video codec: On2 VP6
Video data rate: 400kbps
Audio codec: MPEG Layer III (MP3)
Audio data rate: 96kbps (stereo)

4. Click the Advanced Settings button.

The Advanced Encoding Settings are divided into three distinct areas that can be reached by clicking the appropriate tab. The Encoding tab is where the video is optimized for output. The Cue Points tab is where you can set the timing for other events in the Flash movie based upon a specific point in time in the video (you use the Cue Points in Chapter 9). The Crop and Trim tab allows you to crop a video and set the in and out points by the numbers.



Please select a Flash Video encoding profile:

Flash 8 - Medium Quality (400kbps)

Video encoded for playback with Flash Player 8
Video codec: On2 VP6
Video data rate: 400kbps
Audio codec: MPEG Layer III (MP3)
Audio data rate: 96kbps (stereo)

Output filename:

00:00:00.000

☑ Encode video

Video codec: On2 VP6

□ Encode alpha channel

Quality: Medium

Max data rate: 400 kilobits per second

Frame rate: Same as source fps

□ Resize video

Width:

Height:

pixels

Key frame placement: Automatic

Key frame interval: frames

☑ Maintain aspect ratio

☑ Encode audio

Audio codec: MPEG Layer III (MP3)

Data rate: 96 kbps (stereo)

5. On the Encoding tab, select the On2 VP6 codec to choose the codec to be used for the encoding.

The term codec is an acronym for en**CO**der/**DEC**oder. The codecs used for Flash video are the On2 VP6 codec and Sorenson Spark, which is a lite version of the Sorenson Squeeze codec. The On2 VP6 codec is a new addition to Flash Professional, and if the digital video being encoded contains an alpha channel, it is the one to use. Your choice of codec is up to you because they both produce high-quality video.

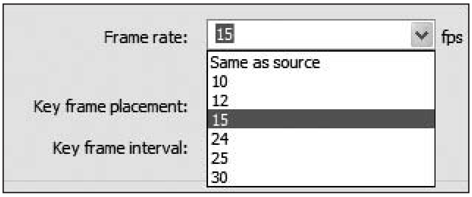
These two codecs are the real workhorses of the video-creation process. They can compress a QuickTime video that is 45 MB and create an FLV file that weighs in around 13 MB or lower. This happens because the codecs see the video as being nothing more than a series of pictures on a Timeline. The frames where the picture is uncompressed are called keyframes and the frames between the keyframes—called difference or delta frames—are where the compression occurs. The more frames there are between the keyframes, the greater the reduction in file size during the compression process. In regular video production, keyframes are usually found on every 15th or 30th frame. These two codecs are designed to produce excellent results for keyframes spread out to every 48th frame, example, which results in a smaller file size.

Another feature of these two codecs that is critical to the use of Flash video. They add metadata, such as the video's width and height, to the FLV file. This metadata is accessible to Flash Player and can be used by ActionScript to manipulate the video.

Tip *If you are working with a video producer or video production company, try to have them give you a digital video file that is uncompressed. If it must be compressed, specify it to be compressed using either the video or animation codec. These two video codecs are lossless, meaning the information in the video remains intact. The two codecs in Flash Professional 8 are lossy, meaning data will be lost during the compression process. The best way of understanding this is to think of JPG image compression. If you recompress a JPG image using the JPG format, the image will actually be lower in quality than what you started with (look for banding in gradients, square blocks, and artifacts in your images). The same principle applies to video. If you compress a video that is already compressed–Sorenson or Cinepac–the quality of the FLV file is lower.*

6. Open the Frame Rate drop-down list and select 15 from the choices.

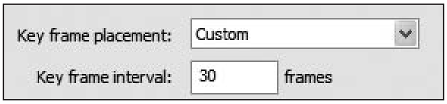
Although Flash animations play quite well at a frame rate of 24 fps, Flash video has its problems at that speed. You also have to understand that frame rate is independent of the host SWF file. If the video has a frame rate of 12 fps and the SWF file where the FLV file is placed has a frame rate of 24 fps, the Flash movie in the Web page will play back at 24 fps, whereas the video in that movie will play back at 12 fps. The most common bit of advice around frame rate is to have the two of them equal to or multiples of each other.



Frame rate: 15 fps

7. Select Custom in the Key Frame Interval drop-down list and enter 30 in the Key Frame Interval input box.

When setting the number of keyframes, pay attention to the video. If there is a lot of motion in the video, such as cars moving, people walking, or moving scenery, use a lower keyframe number. If the video is nothing more than your dog running in a field, a higher keyframe number will work. Keep in mind that if you have a controller that allows the users to move forward or backward in the video (the official term is *seeking*), what the user sees when they seek is the keyframes.

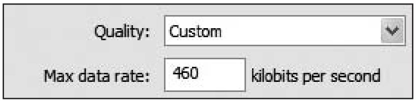


Key frame placement: Custom

Key frame interval: 30 frames

8. Set the Quality to Custom and enter 460 into the Max data rate input box.

If there is one theme running through this part of the lesson, it is this: Data rate controls quality. Bandwidth controls the user experience.



Quality: Custom

Max data rate: 460 kilobits per second

Regardless of the platform used, bandwidth, or the pipe, is the critical consideration. In fact, when deciding to use video you should have a bandwidth strategy in place for the user, the server, and the video. The server bandwidth strategy must revolve around the maximum or total number of users accessing the video. The last thing you need is to have users hanging around waiting for their turn in the line to download and play the video. When it comes to the user, the bandwidth required has to have enough room for the stream and other Internet activities. For example, a user with a 56K dial-up connection can connect and process data at an ideal rate of 56.6 Kbps. In fact, the target data rate for a video in this situation is about 40 Kbps, which leaves some room for other activities on the browser.

A low data rate reduces the number of bits per second shot down the pipe. The result is a lower-quality product that is ideally suited for low bandwidth (dial-up) connections and has a much smaller file size. A higher data rate, such as the traditional target of 350 Kbps for most DSL connections, will result in a great-looking, fast-loading file on a cable modem, but will choke a dial-up connection.

Note *Data rate is simply the amount of data transferred, per second, to the user's computer. This, in turn, determines the bandwidth required to play the video. The data rate calculation is this:*

```
Data rate= w x h x color depth x frames per second/ compression
```

The video for this exercise is 320 x 240 and it uses 24-bit color. If you plug in the numbers and use 60:1 compression, the benchmark for the Spark and On2 VP6 codecs, the data rate for the video at 15 fps is 460,800 bits/sec or 460 kilobits per second, which explains the data rate value you entered.

If you do the calculation for the uncompressed video–no compression number–the calculation gives you a result of 27,648,000 bits per second. To deliver that kind of file, you would need an Internet pipe that is the diameter of a railroad tunnel. Also Flash Player supports a maximum data rate of 4 MB per second; anything over that and you have problems. Toss the compression ratio into the equation and you have a video whose data rate is very comfortably within Player's range.

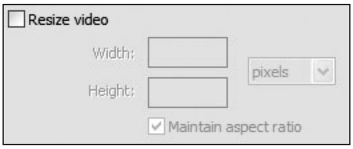
9. Don't resize the video.

If you must resize a video, resize to a lower size. Don't resize to a higher size. What many people tend to forget is that a video is composed of a number of images. If you resize an image, it tends to get fuzzy because an image is composed of a set number of pixels. If you double the size of the image, all you do is make the pixels bigger—the same number of pixels in a larger space—and the result is a fuzzy image.

The standard sizes for video are these:

- 320 x 240
- 240 x 180
- 160 x 120

The first number is easily divisible by 4, and the second is easily divisible by 3. The measurements indicate that the height is 3/4 the width. If you divide the height by the width, you will get 1-1/3 or 4/3. That is the 4:3 ratio. The 4:3 ratio is the standard aspect ratio for digital video and, if you must resize the video, make sure that the Maintain Aspect Ratio radio button is clicked.


☐ Resize video

Width:

Height:

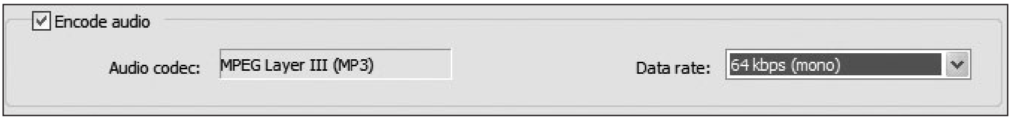
pixels

☑ Maintain aspect ratio

Tip *If your target bandwidth is dial-up, videos that are 240 x 180 or 160 x 120 are the ideal size. All other bandwidth situations should use the 320 x 240 size or smaller.*

10. Click the Encode Audio radio button and select 64 Kbps (mono) from the Data Rate drop-down list.

There are actually two compressor used for FLV file creation: one for audio and another for video.



☑ Encode audio

Audio codec: MPEG Layer III (MP3)

Data rate: 64 kbps (mono)

The choice you make here is rather uncomplicated. These settings will compress the sound portion of the video to the FLV file format, which is an MP3 format. The bit rates presented to you essentially optimize the audio track for FLV file playback. The number chosen has a direct effect upon quality. Higher numbers mean higher quality and less compression. Lower numbers mean lower quality and greater compression.

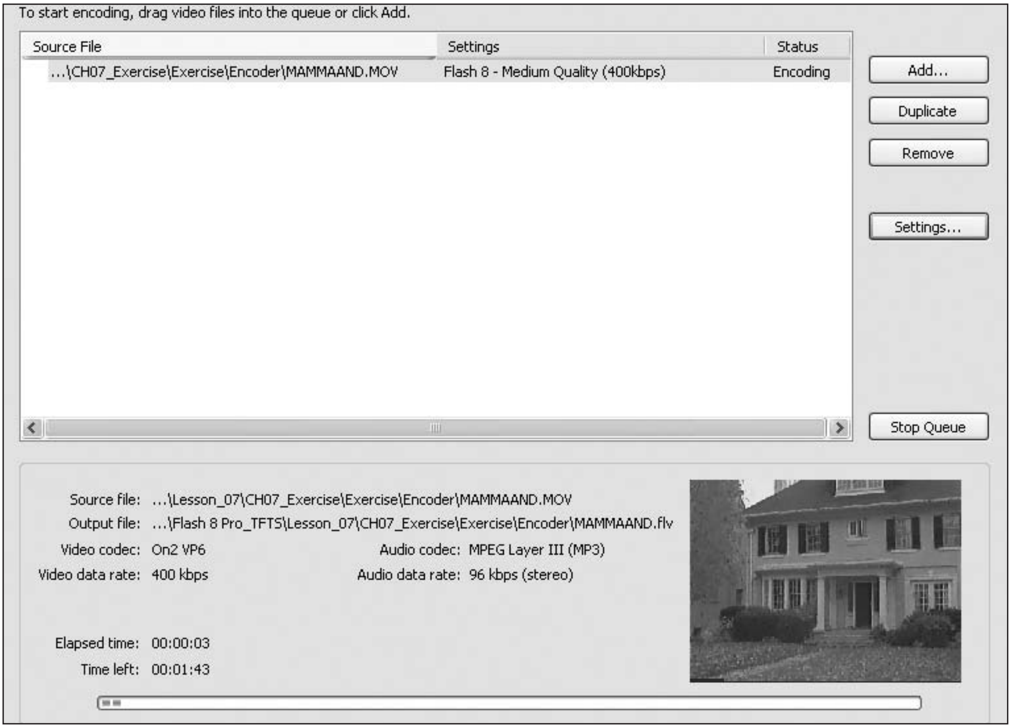
Tip *Choosing mono over stereo will have the effect of reducing the size of the audio track in the video by about 50 percent. Considering the fact the average user does not have a high-end sound system attached to his or her computer, stereo sound can be regarded as a bandwidth-hogging luxury. Use stereo only if your sound requires it as far as user experience goes. If you are working with speech, mono with a lower audio data rate will more than suffice.*

11. Click OK to return to the Video Encoder window; then click the Start Queue button.

The interesting aspect of this window is that you can use it to encode one or a dozen videos. As well, each of the videos can have its own settings.

When the video encodes, you will see a progress bar, and the video will appear during the process. When the video is finished encoding, the resulting FLV file will be placed in the folder containing the original video.

Tip *All videos that you have compressed will be listed in the Encoder dialog box. If you want to redo the compression, simply select the video in the list and click the setting button. To clear the list, select the videos and click the Remove button.*



To start encoding, drag video files into the queue or click Add.

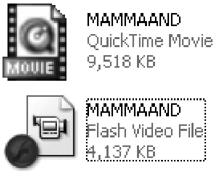
Source File	Settings	Status
...\CH07_Exercise\Exercise\Encoder\MAMMAAND.MOV	Flash 8 - Medium Quality (400kbps)	Encoding

Source file: ...\Lesson_07\CH07_Exercise\Exercise\Encoder\MAMMAAND.MOV
Output file: ...\Flash 8 Pro_TFTS\Lesson_07\CH07_Exercise\Exercise\Encoder\MAMMAAND.flv
Video codec: On2 VP6
Audio codec: MPEG Layer III (MP3)
Video data rate: 400 kbps
Audio data rate: 96 kbps (stereo)

Elapsed time: 00:00:03
Time left: 00:01:43

12. When the video finishes encoding, close the Flash 8 Video Encoder dialog box.

If you want, compare the file size of the FLV file just created to that of the original QuickTime video.



MAMMAAND
QuickTime Movie
9,518 KB

MAMMAAND
Flash Video File
4,137 KB

Creating a Video using the FLVPlayback Component

In the first part of this lesson, you created an FLV file that automatically played through the FLVPlayback component. In this exercise, you link the component to the FLV file.

As you become more comfortable with creating Flash video and adding it to your Web pages, you will see the creation process as essentially involving these two steps:

- Creating the FLV file using the Flash 8 Video Encoder
- Linking the FLV file to a video object, in this case the Player, on the Flash Stage

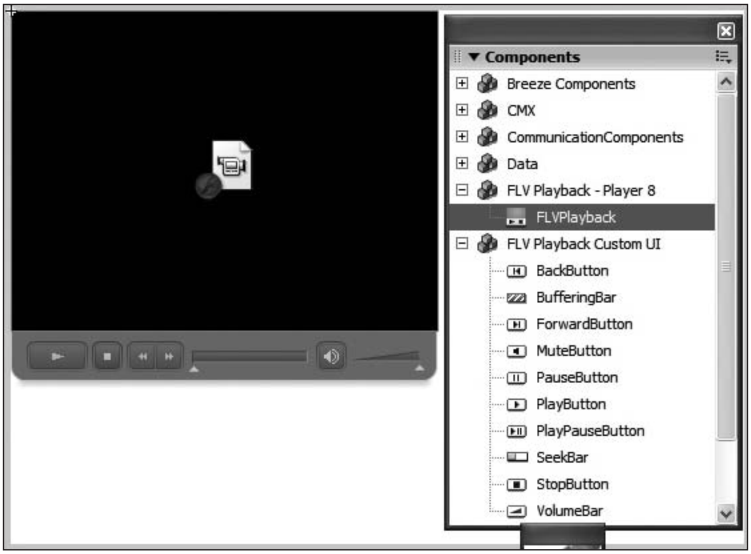
By linking a FLV file to an object, you can reuse the player as the videos change; at the same time you can also customize the Player's properties to enrich the video experience and give the user an even greater degree of control over the video.

1. Open a new Flash document and open the Components panel.

If your Components panel isn't open, you can open it by selecting Window > Components or by pressing the Control+F7 (PC) or Command+F7 (Mac).

2. Double-click the FLVPlayback-Player 8 component and drag a copy of the FLVPlayback component onto the Stage.

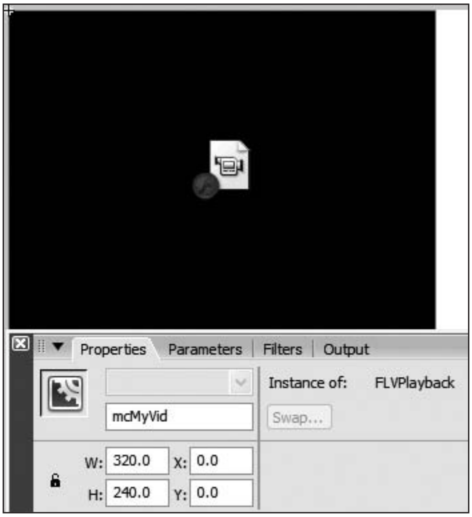
This component is the one automatically added to your movie when you use the Video Import Wizard.



3. Select the component on the Stage, click the Properties tab of the Property inspector, click once in the Instance Name area of the Property inspector, and name the component mcMyVid.

Tip *Although you won't be using the instance name in this exercise, giving components instance names is a good habit to develop.*

4. With the component selected on the Stage, in the Property inspector, set the width to 320, the height to 240 and the *x* and *y* coordinates to 0.



mcMyVid

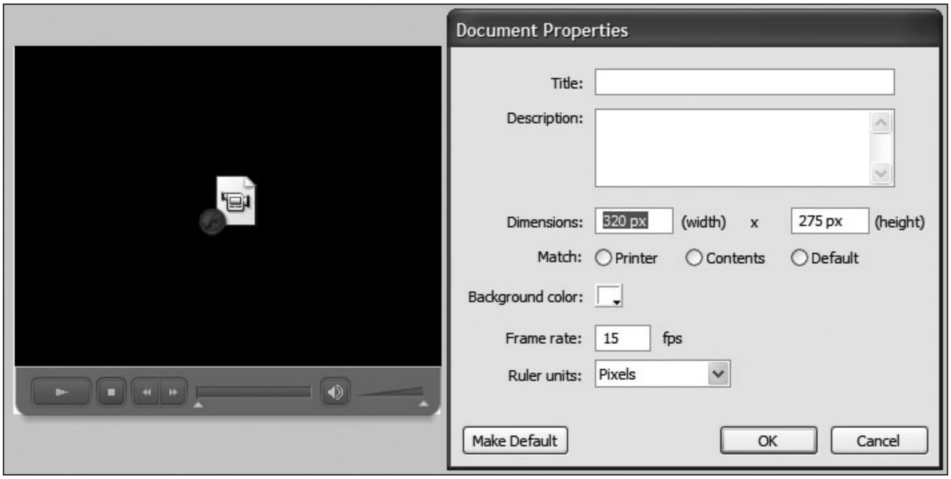
W: 320.0 X: 0.0

H: 240.0 Y: 0.0

The default value for this component is a lot smaller than the dimensions of the FLV file you have created. As well, this will be a video placed into a Web page. Placing the component in the upper-left corner of the Stage ensures that the SWF file will not contain wasted space on the Stage.

5. Click the Stage and then click the Size button on the Property inspector. When the Document Properties dialog box opens, set the Stage dimensions to 320 x 275 and the Frame Rate to 15 fps. Click OK.

The white space under the video area will be used by the Controller you will add to the Component. It requires about 30 pixels of space under the video or there is a risk of it getting "trimmed off" upon playback. The change in the frame is to ensure that the frame rate for the SWF file holding the video has the same frame rate as the FLV file.



Document Properties

Title:

Description:

Dimensions: 320 px (width) x 275 px (height)

Match: ○ Printer ○ Contents ○ Default

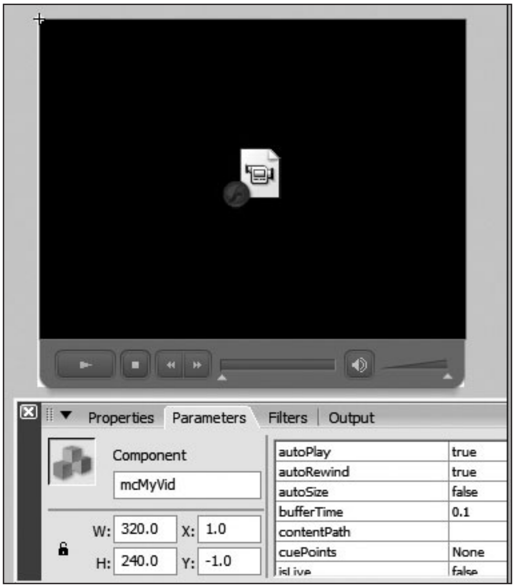
Background color:

Frame rate: 15 fps

Ruler units: Pixels

6. **Click once on the video component on the Stage and click the Parameters tab in the Property inspector.**

All components, this one included, use customizable values that determine how the component looks and often functions. When you select a component, the parameters will appear in two places: the Parameters tab of the Property inspector or in the Parameters window of the Component inspector.



Component

mcMyVid

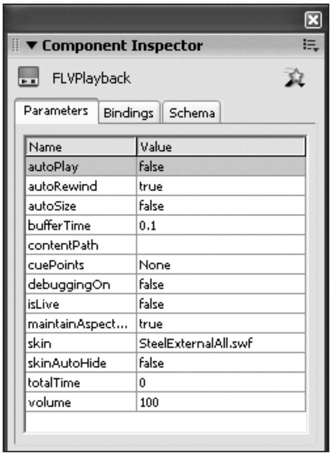
W: 320.0 X: 1.0

H: 240.0 Y: -1.0

autoPlay	true
autoRewind	true
autoSize	false
bufferTime	0.1
contentPath	
cuePoints	None
isLive	false

7. Open the Component inspector and click the Parameters tab to view the full list of parameters for this component.

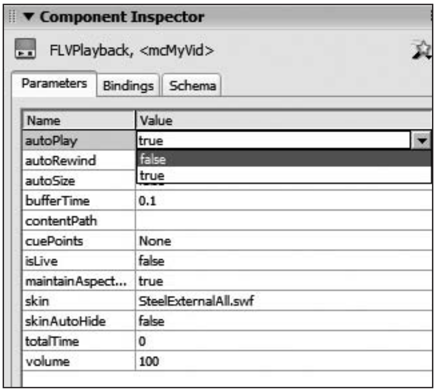
Although you can use the Property inspector, most Flash developers tend to use this panel instead of the Property inspector. To open the Component inspector, you can either select Window > Component Inspector or press Alt+F7 (PC) or Option+F7 (Mac). If you don't see the parameters when you click the Parameters tab in the inspector, select the component on the Stage.



Name	Value
autoPlay	false
autoRewind	true
autoSize	false
bufferTime	0.1
contentPath	
cuePoints	None
debuggingOn	false
isLive	false
maintainAspect...	true
skin	SteelExternalAll.swf
skinAutoHide	false
totalTime	0
volume	100

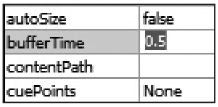
8. Click the Value area for autoPlay and select False from the drop-down list.

This selection means the user will have to click the Play button in the controller when the video loads into the browser to play the video.



Name	Value
autoPlay	true
autoRewind	
autoSize	
bufferTime	0.1
contentPath	
cuePoints	None
isLive	false
maintainAspect...	true
skin	SteelExternalAll.swf
skinAutoHide	false
totalTime	0
volume	100

9. **Click once in the *`bufferTime`* value and enter .5.**

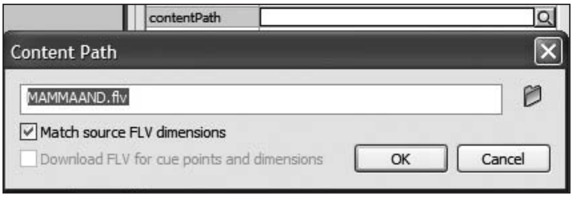


autoSize	false
bufferTime	0.5
contentPath	
cuePoints	None

The value you just entered tells Flash to store one-half second of video in a buffer. This is especially useful when a video streams because a buffer minimizes dropped video frames. When a video plays, the frames must be shown at specific times. If there is a delay, the frames between where the playback head is currently located and where it should be are dropped. The result is a "jerky" video. Creating a buffer and storing frames in the buffer is a great habit to develop. You can use any value you wish but the more common values are between .5 and 2 seconds. Remember the more you buffer, the longer the wait for the video to start.

10. **Click once in the *`contentPath`* value area and enter the name of the FLV file.**

Values for the path can be either Absolute or Relative. For example, if the final SWF file and FLV file are to be located in the same directory on the server, a relative path—the name of the FLV file—will suffice. If you have the FLV files located in a separate folder, you would enter `./flvFolderName/VideoName.flv`.



contentPath

Content Path

MAMMAAND.flv

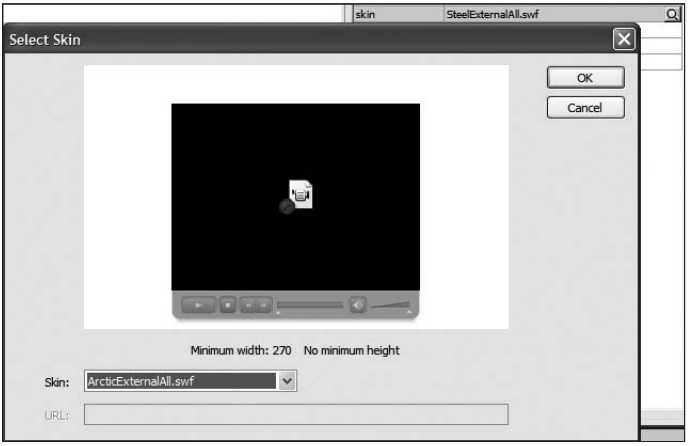
☑ Match source FLV dimensions

□ Download FLV for cue points and dimensions

Tip *You can double click the Browse Button in the content Path and navigate to the folder containing the FLV.*

11. **Click once on the Skin value and then click on the magnifying glass in the value input area to open the Select Skin dialog box.**

This is the same dialog box used in the wizard.



skin SteelExternalAll.swf

Minimum width: 270 No minimum height

Skin: ArcticExternalAll.swf

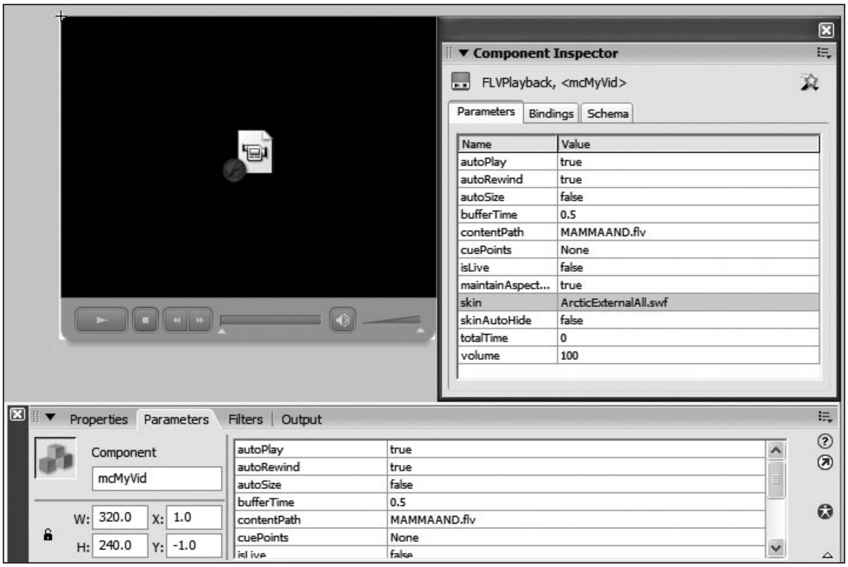
URL:

12. Select `ArcticExernalAll.swf` from the list of skins and click OK.

The skin selected will appear in the Component inspector.

13. Close the Component inspector.

If you click the Parameters tab of the Property inspector, the values chosen in the Component inspector appear here as well.



FLVPlayback, <mcMyVid>

Name	Value
autoPlay	true
autoRewind	true
autoSize	false
bufferTime	0.5
contentPath	MAMMAAND.flv
cuePoints	None
isLive	false
maintainAspect...	true
skin	ArcticExternalAll.swf
skinAutoHide	false
totalTime	0
volume	100

Component

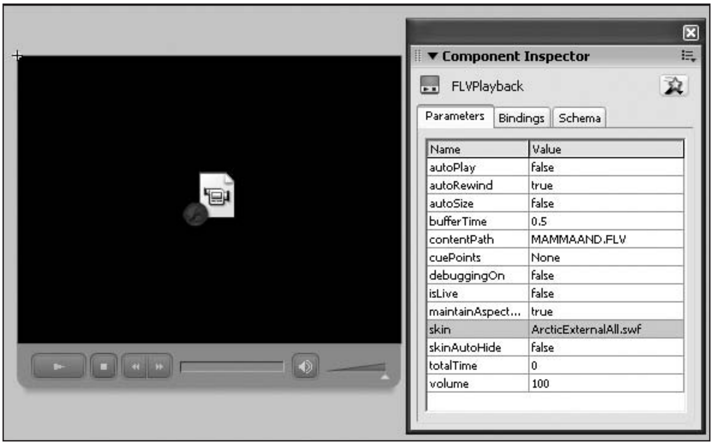
mcMyVid

W: 320.0 X: 1.0

H: 240.0 Y: -1.0

autoPlay	true
autoRewind	true
autoSize	false
bufferTime	0.5
contentPath	MAMMAAND.flv
cuePoints	None
isLive	false

14. **Save the Flash file to the same folder as the FLV file and test the movie.**

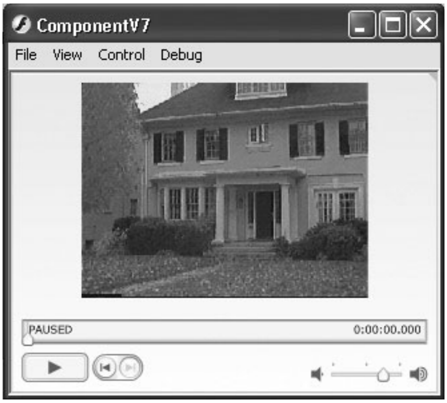


Name	Value
autoPlay	false
autoRewind	true
autoSize	false
bufferTime	0.5
contentPath	MAMMAAND.FLV
cuePoints	None
debuggingOn	false
isLive	false
maintainAspect...	true
skin	ArcticExternalAll.swf
skinAutoHide	false
totalTime	0
volume	100

Creating a Video using the MediaPlayback Component for Flash Player 6/7

The inclusion of the MediaPlayback component in Flash Professional 8 is an acknowledgment that at least for the first year of its existence, not all Web users will have the Flash Player 8 installed on their computers. This could be because of corporate IT policies, where users are not permitted to add or upgrade software on their company computer.

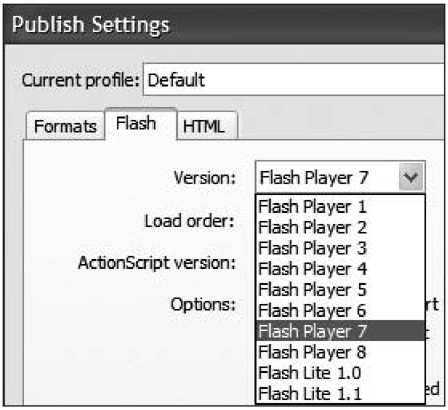
In this case, a previous version of Flash Player, version 6 or version 7, will have to be used for playback. This will be accomplished through the use of the MediaPlayback component that was included in Flash MX 2004 Professional and is also a part of the Component panel in Flash Professional 8.



1. **Open a new Flash movie by choosing File › New and then save it to the same location as your FLV file.**

This ensures that the FLV file will be found when it is needed at playback.

2. **Select File › Publish Settings. Click the Flash tab and select Flash Player 7 from the Version drop-down menu. Click OK to close Publish Settings.**



Publish Settings

Current profile: Default

Version: Flash Player 7

Load order:

ActionScript version:

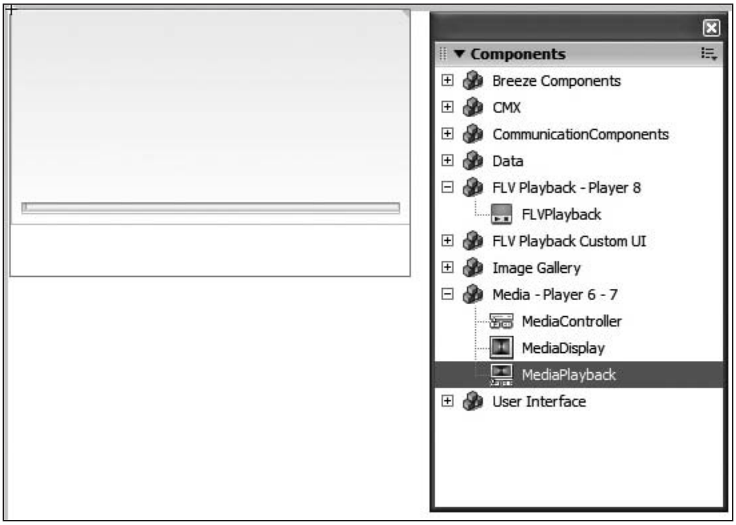
Options:

By selecting the Player even before the file is constructed, you ensure that the file will play properly in Flash Player 7.

Tip *The Flash Player 8 video components are clearly named as such in the Component panel for a reason. They are not compatible with earlier versions of Flash Player. Thankfully, you don't discover this fact when the video is uploaded to the server and played through a browser. You will be notified of this, through a warning box, when you test the movie.*

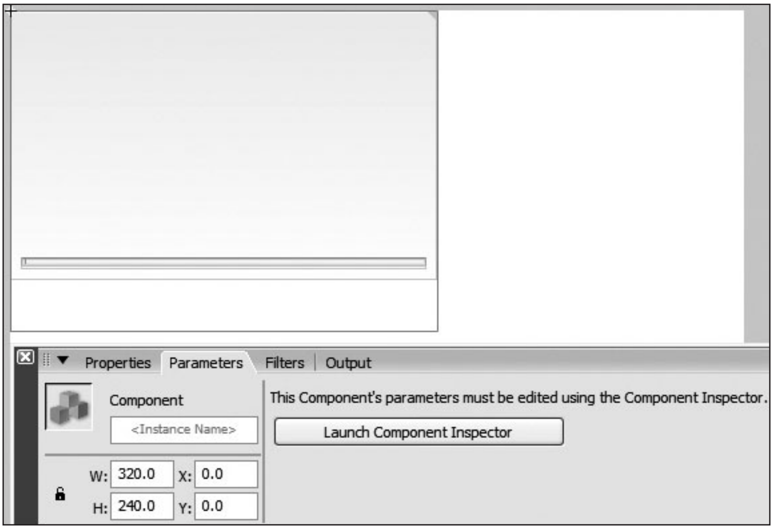
3. **Open the Components panel and drag a copy of the MediaPlayback component from the Media-Player 6-7 Component group to the Stage.**

The MediaPlayback component is a composite of the two other components in the group: MediaDisplay and MediaController.



4. **With the component selected on the Stage, set the width and height of the component to 320 x 240 set the X and Y locations to 0,0. Click the Parameters tab and click the Launch Component Inspector button.**

When you drag the component onto the Stage and click the Parameters tab in the Property inspector, the parameters are replaced with a Launch Component Inspector button. This difference is the way the media Component parameters were determined in previous versions of Flash.

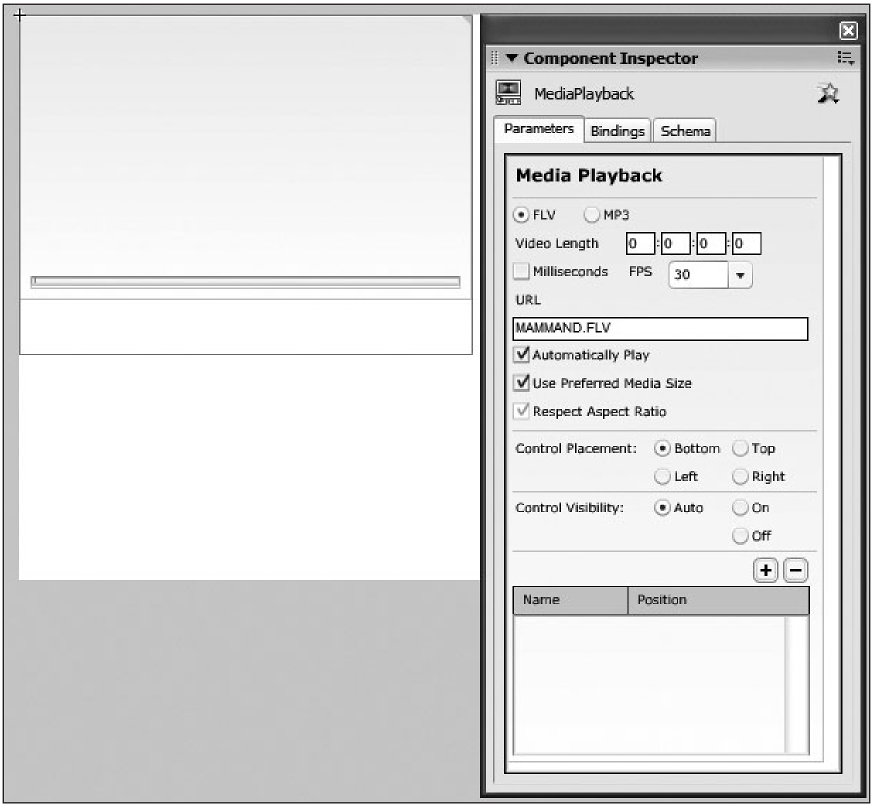


This Component's parameters must be edited using the Component Inspector.

W: 320.0 X: 0.0
H: 240.0 Y: 0.0

5. When the Component inspector opens, select FLV, enter the name of the FLV file, and deselect Automatically Play.

This tells the component that you are using a FLV file. As in the previous exercise, the name of the component is actually the path to the FLV file. Deselecting Automatically Play lets the user decide whether or not to view the video. The MP3 button is used in Flash MX 2004 for MP3 files.



Component Inspector

MediaPlayback

Media Playback

● FLV ○ MP3

Video Length 0 : 0 : 0 : 0

□ Milliseconds FPS 30

URL

MAMMAND.FLV

☑ Automatically Play

☑ Use Preferred Media Size

☑ Respect Aspect Ratio

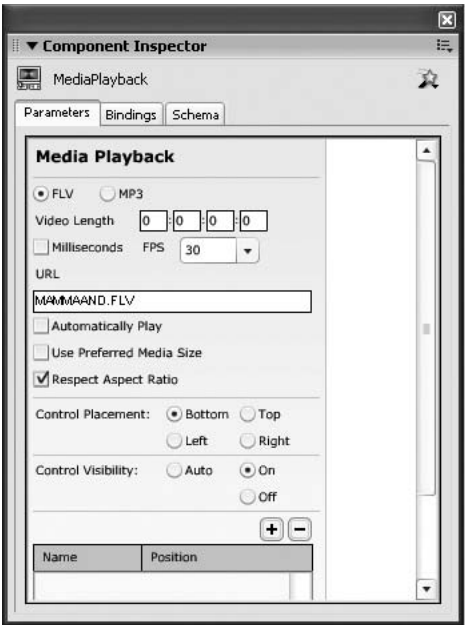
Control Placement: ● Bottom ○ Top ○ Left ○ Right

Control Visibility: ● Auto ○ On ○ Off

Name	Position

6. Select Bottom in the Control Placement area and select On for Control Visibility. Close the Component inspector.

When you choose a Control Placement location, the choice made will be immediately reflected in the component on the Stage. Selecting Auto for Control Visibility means the user has to roll over the video to see the playback controls.



Media Playback

● FLV ○ MP3

Video Length 0 : 0 : 0 : 0

□ Milliseconds FPS 30

URL

MAMMAAND.FLV

□ Automatically Play

□ Use Preferred Media Size

☑ Respect Aspect Ratio

Control Placement: ● Bottom ○ Top ○ Left ○ Right

Control Visibility: ○ Auto ● On ○ Off

Name	Position

Tip *If you deselect Autoplay and turn off Control Visibility in this component or the Flash Player 8 components, be aware that the user won't be able to play the video. The video will be there, but the user will have no way of playing it.*

7. Save and test the movie.

Using the FLV Custom UI Components to Control the FLVPlayback Component

There will be occasions when the user will be given limited control over video playback. This could include, for example, the use of a video in an e-learning situation, in which the student simply has to watch a video. In this scenario, the only two controls needed would be a play button to start the video and a pause button to stop it from playing. This is the purpose of the FLV Custom UI components.

This collection of 10 components, new to Flash Professional 8, lets you build your own video controller and wire it up using ActionScript. In essence, each one is a movie clip or button and, as such, they can be customized.

When one of these components is added to the Stage, not only is the component added to the library but the component is also added to the FLVPlayback Skins folder. You could, for example, change the color of the gradient used in the Play button or even change the green in the PlayButtonOver to blue. Best of all, these changes don't affect the original component in the Component panel.

1. Open a new Flash movie by choosing File › New and then add two new layers above the existing layer Name your three layers Video, Buttons, and Actions.

2. Select the Video layer and drag a copy of the FLVPlayback component to the Stage.

Select the Playback component and, in the Property inspector, add these values:

- Instance name: `myVid`
- W: 320
- H: 240
- X: 0
- Y: 0

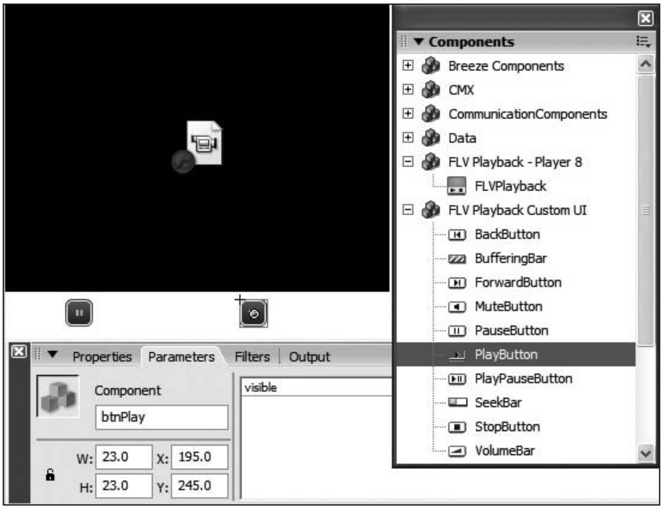
3. Select the Buttons layer and open the FLV Custom UI-Player 8 components. Drag a copy of the PlayButton and the PauseButton components to the Stage.

With buttons on the Stage, select the Pause button and add these values in the Property inspector:

- Instance Name: `btnPause`
- X: 50
- Y: 245

Select the Play button and add the following values in the Property inspector:

- Instance Name: `btnPlay`
- X: 195
- Y: 245

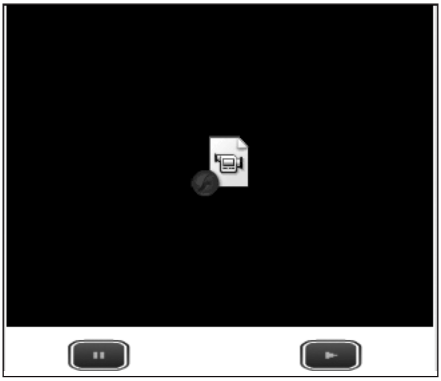


Component
btnPlay
W: 23.0 X: 195.0
H: 23.0 Y: 245.0
visible

Tip *You don't necessarily have to drag components onto the Stage to get them to appear in the library. You can drag them from the Components panel directly into the Library panel.*

4. Click once on the Stage and, in the Property inspector, click the Size button.

When the Document Properties dialog box opens, change the Stage size to 320 pixels wide by 275 pixels high. Click OK to accept the change and close the Document Properties dialog box.

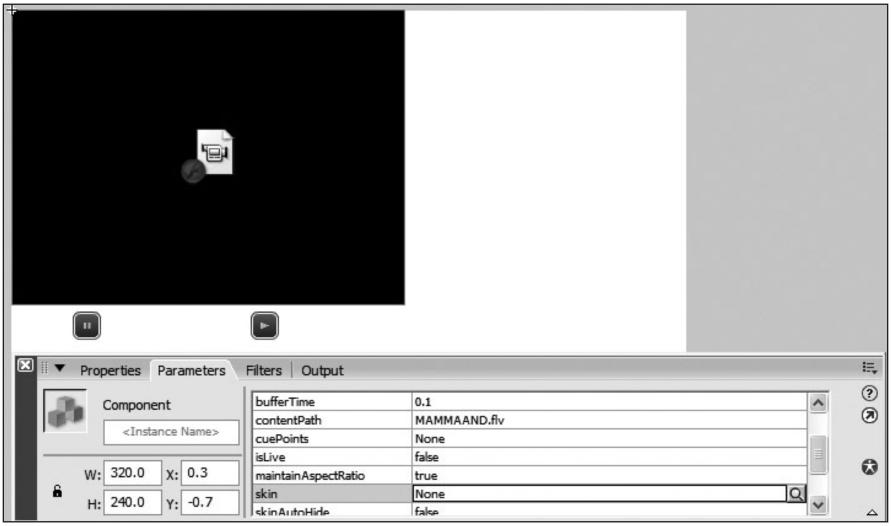


5. Select the FLVPlayback component and click the Parameters tab in the Property inspector.

Use the following values in the Parameters area:

- autoPlay: False
- contentPath: `MAMMAAND.FLV`
- Skin: None

The Play button will control the playback of the movie and because you have two buttons, there is no need for a skin.



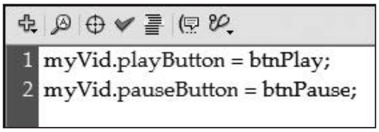
W: 320.0 X: 0.3
H: 240.0 Y: -0.7

bufferTime	0.1
contentPath	MAMMAAND.flv
cuePoints	None
isLive	false
maintainAspectRatio	true
skin	None
skinAutoHide	false

6. Select the Actions layer and press F9 to open the Actions panel. Enter the following code into the panel:

```
myVid.playButton = Play btn;
myVid.pauseButton = btnPause;
```

By setting the `autoStart` parameter of the FLVPlayback component to `false`, you have essentially stopped the movie from playing. The FLV file is streamed into the FLVPlayback component, which means it can be controlled by the `play()` and `pause()` methods contained in the button components. In this case, when the Play button is released the stream starts because of the `play()` method.



```
myVid.playButton = btnPlay;
myVid.pauseButton = btnPause;
```

The play and pause buttons are simple buttons and nothing more. Because they inherit the component architecture, the video component knows all it needs to know about a button.

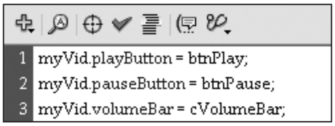
Once it finds out there is a button attached to it (the `myVid.playButton`), it latches on to it by adding itself to the button's list of listeners. Once the button is clicked, the FLVPlayback component is notified and will take action based on where you assigned the button (playButton, pauseButton, and so on). The button itself doesn't have any function; it simply tells the FLVPlayback component that it has been clicked. The component does the rest of the work.

7. Drag a copy of the VolumeBar component onto the Stage and give it the instance name *`cVolumeBar`*.

The volume bar is treated in exactly the same manner as the two buttons on the Stage. It is a component with methods, properties, and events that are accessed by the FLVPlayBack component.

8. Open the Actions panel and add the following line of code:

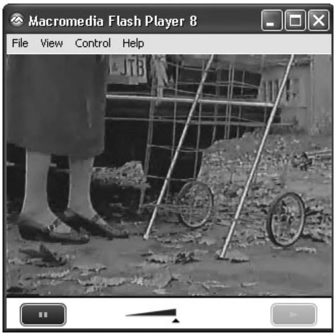
```
myVid.volumeBar = cVolumeBar;
```



```
myVid.playButton = btnPlay;
myVid.pauseButton = btnPause;
myVid.volumeBar = cVolumeBar;
```

9. Save the movie and test it.

The movie shows you only the first frame of the video. If you click the Play button, the video plays. Click the Pause button, and it stops. If you click the Play button once more, the video resumes playing. Drag the volume controller, and the volume of the sound playing from the video increases or decreases depending on which direction you dragged the mouse.



Note *Now that you understand how to add the UI components to a movie and control the video play back, here is how the rest of them are connected. The syntax is:*

```
Instancenameof FLVPlaybackcomponent.UIcomponentName =
instancenameofUIcomponent
```

In the following examples, the instance name for the FLVPlayback component is `"myFLVPLayback"`.

```
myFLVPlayback.playButton = myPlayButton;

myFLVPlayback.pauseButton = myPauseButton;

myFLVPlayback.playPauseButton = myPlayPauseButton;

myFLVPlayback.stopButton = myStopButton;

myFLVPlayback.backButton = myBackButton;

myFLVPlayback.forwardButton = myForwardButton;

myFLVPlayback.muteButton = myMuteButton;

myFLVPlayback.volumeBar = myVolumeBar;

myFLVPlayback.bufferingBar = myBufferingBar;

myFLVPlayback.seekBar = mySeekgBar;
```

What You Have Learned

In this lesson you have:

- Created an FLV file using the Video Wizard (pages 255–263)
- Created an FLV file using the Flash 8 Video Encoder (pages 263–271)
- Optimized a video for Web playback using the advanced settings of the Video Encoder (pages 263–271)
- Added a skin to a Flash Professional 8 FLVPlayBack component (pages 271–277)
- Linked an FLV file to a component (pages 271–277)
- Set the parameters of the FLVPlayBack component (pages 271–277)
- Used the Flash 6/7 MediaPlayback component to play a video through a previous version of Flash Player (pages 277–281)
- Used the FLV UI components to control the video playing in the FLVPlayback component (pages 281–286)