Creating a Video Player in Flash Professional 8

By: Tom Green

In the first part (http://www.communitymx.com/abstract.cfm?cid=553E4) of this overview of the new video features of Flash Professional 8, I showed you how to use the new tools in Flash Professional 8 — The Wizard and the Flash 8 Video Encoder — to create the FLV file. In this installment I will walk you through the use of the new **FLVPlayback** component and the new **FLV Custom PlayBack UI** components.

Though the Wizard will create the Player and place it on the stage, you can also create the same Player without the use of the Wizard. This is because there will be occasions where you are simply handed the FLV and instructed, "Get it playing." Here's how:

1. Open a new Flash Professional 8 document and if the Components panel is not visible select **Window > Components**.

   When the Panel opens you will notice there are now three component sets aimed at FLV Playback. They are:

   - **FLVPLayback Player 8**: This component can only be used if the Flash 8 Player is being used for video delivery.
   - **FLV Playback Custom UI:** This series of controls can be used when you don't add the controller to the Playback component. They only work in Flash Player 8.
   - **Media Player 6-7 :** Select these components if Flash Player 7 or 6 is your player of choice.

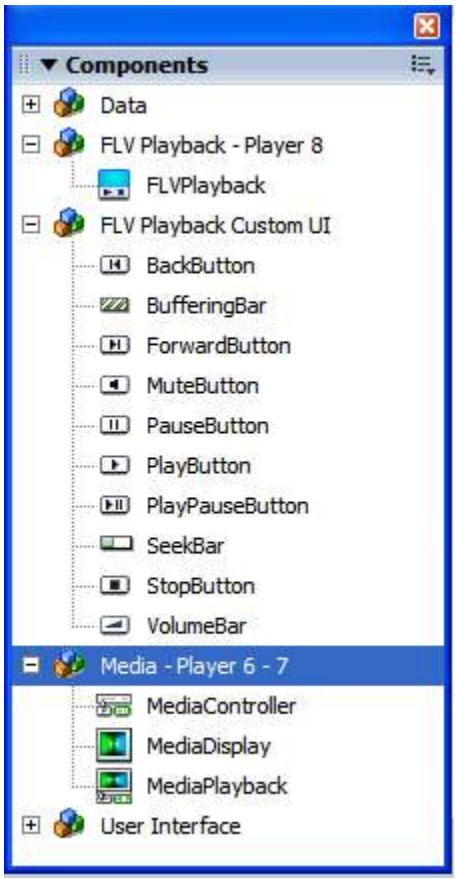

Image 1: *The video components now target a player version.*

2. Drag a copy of the FLVPlayback component from the FLV Playback-Player 8 group to the stage.

   The first thing you will notice is the component contains both an FLV icon and a controller. Though not necessary for this exercise, select the component on the stage and give it the instance name of **mcMyVideo**. The other thing to check on the Property inspector is the dimensions of the video player. The video we will be using is 320 X 240 and the Width and Height measurements for the component should match the video's dimensions.

3. With the video component selected, click once on the Parameters tab of the Property inspector or select **Window > Component Inspector** to open the inspector for the FLVPlayback component.

   Either selection works although I prefer to use the Component inspector because everything is visible in one compact area.

   Use the following values:

   - **AutoPlay = False.** This lets the user start the video.
   - **bufferTime= 0.5.** This tells Flash how much video to place in a buffer to ensure smooth playback.
   - **contentPath:** Click the folder and navigate to the MAMMAAND.FLV file created in the previous article or use the one included in this article's download.
   - **skinAutoHide= false.** This ensures the controls are always visible.

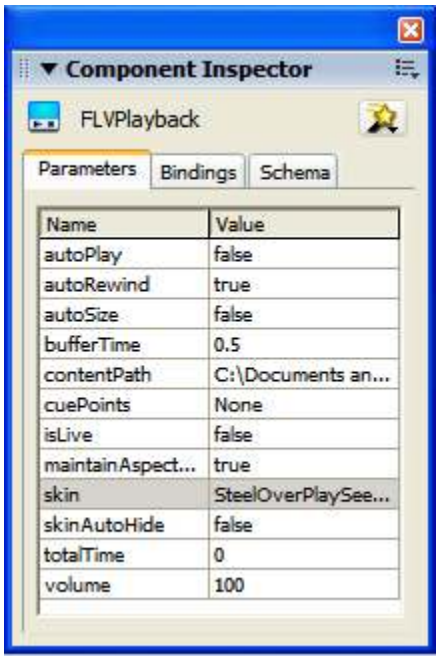

Image 2: *Use either the Component Inspector or the Property inspector to set the FLVPlayback component's parameters.*

4. Click once in the Skin area and then click the Magnifying glass to open the Select Skin dialog box.

   All 32 of the player skins are available to you including any custom skins you may have created.

5. Select **MojaveExternalAll.swf** from the skins menu and click OK.

   This selects a brownish set of controls that sit under the video.The interesting aspect of this dialog box is that you get to see the control styles and their location in the Preview. When you click OK, the dialog box closes and the controller is added to the FLVPlayback component on the stage.

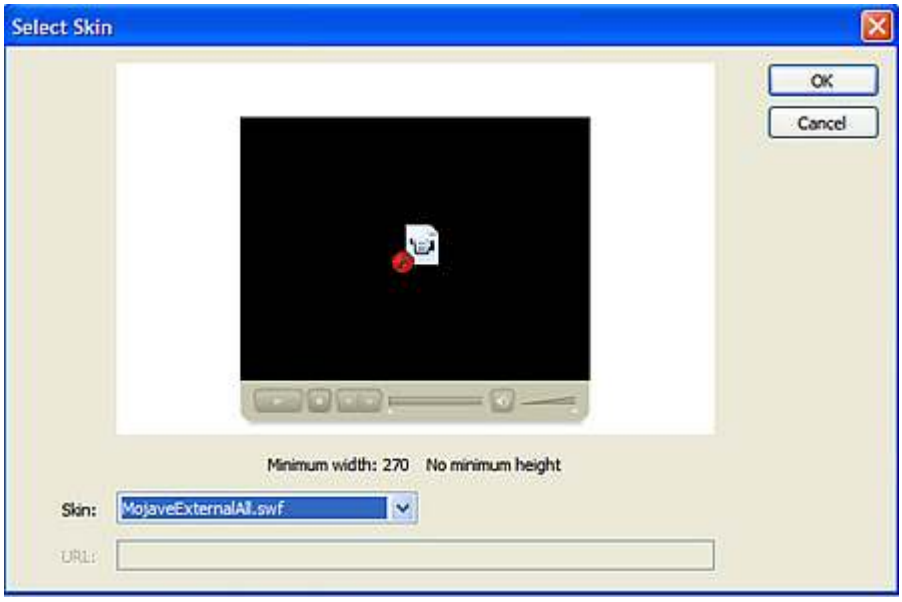

Image 3: *Use the Select Skin dialog box to preview you controller style and placement.*

6. Save the movie and test the file by pressing Control + Enter (PC) or Command + Return (Mac).

Using the FLV Custom UI Components to Control Playback.

There will be occasions where a full bore set of controls isn't necessary. In this case, the UI controls included with Flash Professional 8 will solve that issue for you. This set of 10 components let you build your own video controller and they are dead simple to "wire up". As well, because each one is a movie clip, they can be customized.

Here's how to build a custom controller:

1. Open a new Flash movie and add two layers to the timeline. name the three layers **video**, **buttons** and **actions**.
2. Select the video layer and drag a copy of the FLVPlayback component to the stage. With the component selected give it the instance name of **myVideo** in the Property inspector.
3. Click the parameters tab in the Property inspector and use the following settings:
   - **AutoPlay**= False
   - **Skin**= none
   - **contentPath**= MAMMAAND.FLV. You can also navigate to your own FLV if you don't want to use the one included in the Download.
4. Select the Buttons layer and open the FLV Custom UI- Player 8 components. Drag a copy of the **Play** button and the **Pause** button to the stage.
5. Select each button and give them instance names of **Play_btn** and **Pause_btn** in the Property inspector.

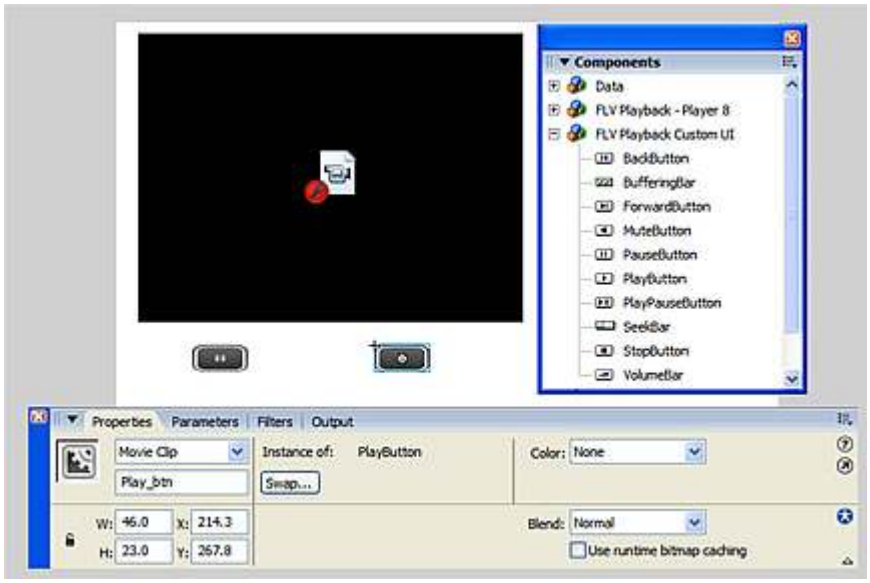

Image 4: *The buttons are added to the stage and given instance names*,

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

```
myVideo.playButton= Play_btn;
myVideo.pauseButton= Pause_btn;
```

7. Save and test the movie.

By setting the autoStart parameter of the FLVPlayback component to false you stopped the movie from playing when it loads. The FLV file is streamed into the FLVPlayback which means it can be controlled

by the `play()` and `pause()` methods contained in the button components. In many respects those two buttons on the stage are nothing more than simple buttons. As such they inherit the component architecture and the video component knows all it needs to know about the play and the pause buttons.

Once the FLVPlayback component finds out there is a button attached to it — **myVideo.playButton —** it "latches" on to it by adding itself to the button's list of listeners. Press the button and the FLVPlayback component is notified and will take the action based on which FLV Custom UI component is assigned to the the player. The button doesn't do anything other than tell the video component it has been clicked. The video component is what is doing all of the work.

Let's add to the functionality of this player by giving the user the ability to control the video's volume.

1. Drag a copy of the VolumeBar component to the stage and give it the instance name of **cVolumeBar**.
2. Open the Actions panel and add the following line to line 3 of the code:

```
myVideo.volumeBar = cVolumeBar;
```

3. Save and test the video.

Now that you have seen how easy it is to add a playback control to manage video here is the syntax to be used for the UI components:

- **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.bufferingyBar = myBufferBar;**
- **myFLVPlayBack.seekBar = mySeekBar;**

The seek bar only works if there are cue points in the FLV or the video is being streamed from a Flash Communication server.

The FLV Playback Custom UI components only work with Flash Player 8. Using them in a player targeted at Flash Player 6 or 7 will result in an error message.

Image 5: *The controls are in place and the video is playing.*

Conclusion

As you have seen from this article, the process of adding, creating and controlling Flash video is now an absolute breeze in Flash Professional 8. The FLVPlayback component is a major improvement over the previous iterations of the component and the ability to simply change the parameters and add a skin to the component are items the developer community will welcome.

You also saw how easy it is to create your own controller using the FLV Playback Custom UI components.

It may just be that we are in for a video revolution on the web.

Approximate download size: 4.2MB

Keywords
Flash 8, Flash Professioal 8, Flash Video, Flash Video Encoder. Flash Video Encoder 8, Video, Components, Flash Video Components.