Flash Article

Flash Video Template: Video Presentation with Navigation

Dan Carr

www.dancarrdesign.com

This video presentation Flash template is great for basic video presentations, such as when you want the user to see the main topics of a video and be able to navigate quickly to them. It does not contain some of the more sophisticated features such as synchronizing content on the screen with the video playback, as described in my other tutorial, Flash Video Template: Spokesperson Presentation with Synchronized Graphics. However, you should have little trouble modifying it to work well for your own needs.

Preview the Video Presentation with Navigation template

This is the second version of the template updated to work with the new Flash Professional 8 FLVPlayback component and the FLVPlayback behaviors that I designed. The updated template is built on a standard Flash timeline and has been simplified for easier editing and customization. The template features a video and navigation buttons, which allow the user to jump to topics within the video.

Consider applying the following special touches to bring this presentation to life in your company:

- Add your own video, resize it, and then position it anywhere on the page
- Add your company graphics and colors to any part of the presentation, including the background
- Update the text, graphics, or colors on any button
- Move the navigation bar to the top, sides—anywhere you like

You can easily change this template to fit your needs.

This article describes how to modify all the basic characteristics so your project looks as you want. For example, you will certainly want to add your own video to this presentation. After that, you can update the button text along the left side of the presentation. Finally, you can change the color schemes and logos to match your own.

Note: If you are still using Flash MX Professional 2004 and not Flash Professional 8, please see the previous version of this article.

Requirements

To complete this tutorial you will need to install the following software and files:

Flash Professional 8

Try Buy

Sample files:

vidtemplate_mediapreso.zip (ZIP, 7.9 MB)

Prerequisite Knowledge

This article is geared towards beginning Flash users and developers who are interested in using a simple video navigation template. While programming experience is not necessary to understand the tutorial, you may find it beneficial to review the Flash Video Learning Guide.

Understanding the Project Layout

In this version of the template, the content is distributed on layers on the main Timeline. This simplifies things and provides for easy access to all the editable features in a single location. Take a quick look at the main Timeline of the template shown in Figure 1.

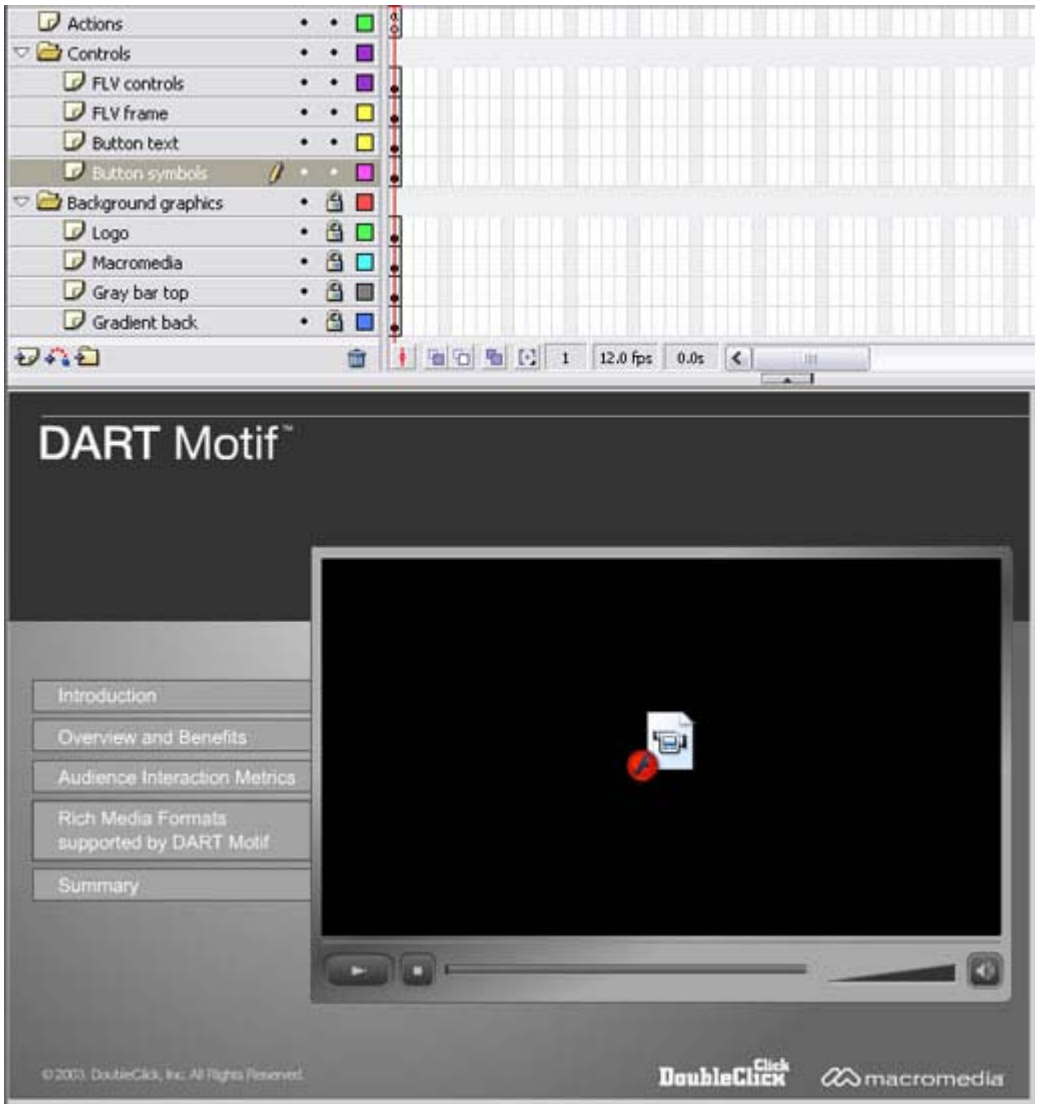

Figure 1. Layers on the main Timeline of the template file

The first thing to notice is that the main Timeline is divided into sections, at least conceptually, by the two layer folders. The Controls folder contains the video controls, video background graphic, and navigation buttons. You will edit these items to change the video and look and feel of the controls. The Background graphics folder contains the

logos and background graphics, which make up the general look and feel of the template. You will edit these items to add your own logo and company branding. The Actions layer contains a few lines of ActionScript code, which you don't necessarily need to edit.

In addition to clearly laying out the pieces of the template, the layer folders enable you to lock or hide easily the sections of the template that you are not currently working on.

Exploring the Symbols in the Library

While you're browsing through the file, open the Library panel (Window > Library) and take a look at its contents (see Figure 2).

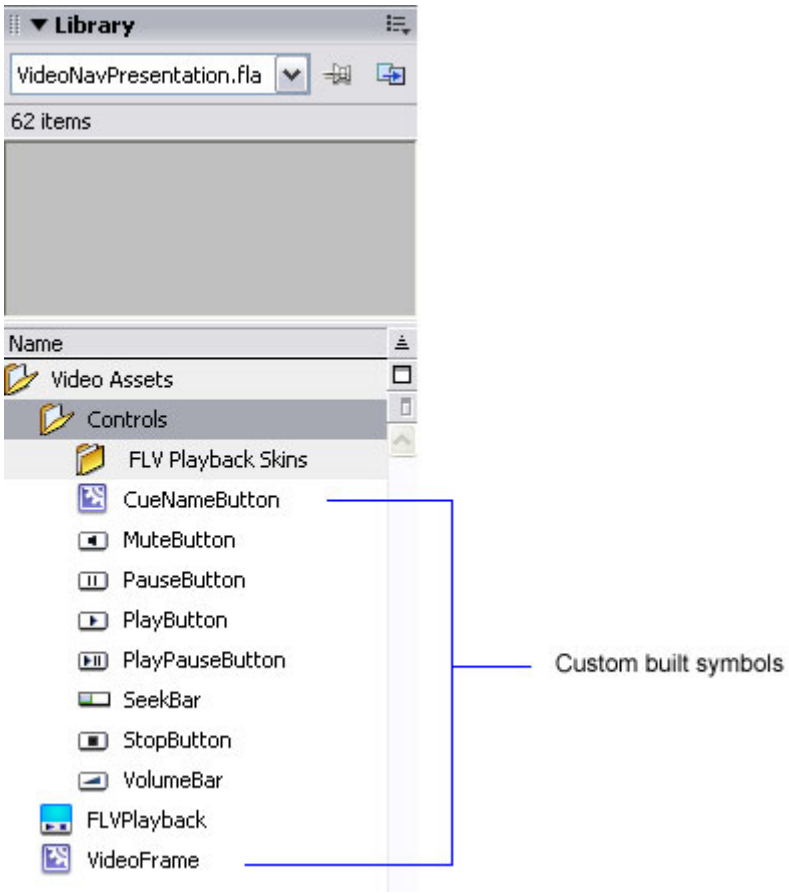

Figure 2. Assets in the template's document library

The first thing to notice is that there are only two custom-built symbols in the Library. The first is the CueNameButton movie clip in the Controls folder. This is used for the navigation buttons, which sit to the left of the video display. The second is the VideoFrame movie clip, which creates the background for the video display. All of the other symbols are gathered from the Flash Professional 8 Components panel in the new FLV Playback categories.

Note: The FLV Playback Skins folder contains customizable skin graphics for each of the video controls.

Exploring the Flash Professional 8 Components Panel

The last stop on the project overview is the Components panel. One of the major upgrades in Flash Professional 8 is the improved video features. The Components panel has been expanded to contain two new categories: FLV Playback – Player 8 and FLV Playback Custom UI (see Figure 3).

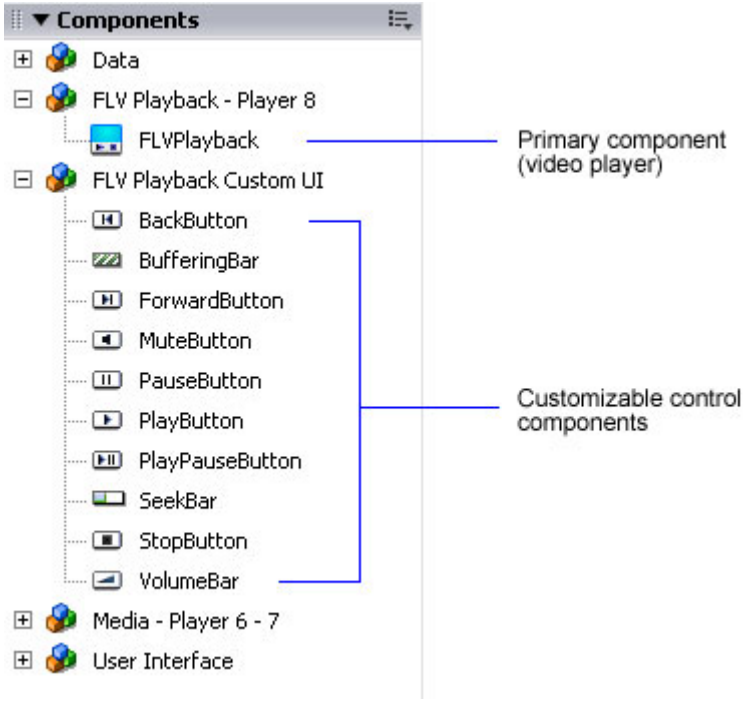

Figure 3. Video controls in the Flash Professional 8 Component panel

Open the Components panel (Window > Components) and take a look at the two new video categories. The Flash team put a lot of effort into making the new FLVPlayback components easy to edit and customize. All of the controls in the template come from the Components panel except for the navigation button, which I designed to complement this set.

When comparing the Components panel to the Library, note that the template does not use every video control available. You will be able to add and remove controls as desired.

Customizing the Video

The FLVPlayback component displays the video in the template (see Figure 4). You will select the FLVPlayback instance to change the layout and parameters of the video. You can change the following items to customize the display:

Video file and video parameters
Size of the video (either automatically or manually)
Location of the video

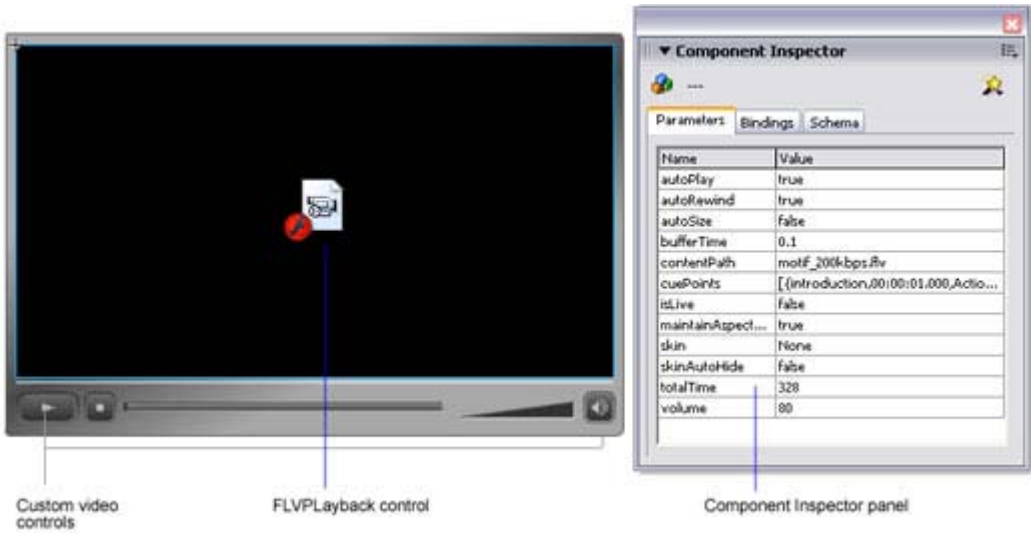

Figure 4. FLVPlayback component instance and its parameters in the Component Inspector panel

The video component in Figure 4 appears as a black rectangle with the Flash video icon in the center. When you select it, it appears with a blue bounding box surrounding the instance. The parameters for the video component can be set in the Property Inspector panel or in the Component Inspector panel. Also notice that the custom video controls (play, stop, seek bar, volume bar, and mute button) are separate controls, which are not attached to the FLVPlayback instance (you will explore changes to these controls in the section "Modifying the Buttons and Video Controls").

Changing the Video and Video Parameters

To change the video and FLVPlayback parameters:

1. Open the VideoNavPresentation.fla file in Flash Professional 8.
2. Click the Flash video icon in the center of the black rectangle to select the FLVPlayback instance. Notice that the FLV Controls layer is highlighted on the Timeline.
3. With the video component still selected, open the Component Inspector panel (Windows > Component Inspector).
4. To change the video, click the `contentPath` field to select it and then click it again to launch the Content Path dialog box (see Figure 5).

Figure 5. Content Path dialog box

5. Type in the path to the FLV file that you want to view or click the folder icon in the top right of the dialog box to browse for the file. Usually you will store the FLV file in a relative path near the Flash movie that displays it. However, if you are streaming the video using Flash Media Server, you will enter an absolute path to the file on the server.

   Notice that you have an option to match the FLV dimensions. This is a great new feature that lets the video component size itself to the target video. If you enter an absolute path into the field, the Download FLV for Cue Points and Dimensions check box becomes active.

6. Enter a path in the text box and choose the Match Source FLV Dimensions check box if desired. Click OK when you're finished.
7. Export the SWF file (Control > Test Movie) to play your video.
8. Return to the FLA file and change the other video parameters as desired. Notice that the `totalTime` parameter is set to 328 to mark the length of the video in seconds. The video used in this sample is an older FLV file, where the `totalTime` value is not embedded in the FLV file's metadata. Entering the length manually enables the seek bar control to work properly.

   If you are using a newer FLV file, then you can remove this setting. Feel free to change it to match the correct length of your video in seconds. If you're streaming the FLV file from Flash Media Server, you will want to turn the `isLive` parameter to `true` and adjust the `bufferTime` parameter to match an appropriate buffer time for your target bandwidth. In this template the `skin` and `skinAutoHide` parameters are not being used.

   For more information on FLVPlayback component parameters, refer to the Flash 8 LiveDocs (Components Language Reference > FLVPlayback Component (Flash Professional Only) > Using the FLVPlayback Component > FLVPlayback Component Parameters).

Note that the FLVPlayback component uses embedded metadata in the FLV file to determine the size and duration of the video for automated features such as "Match source FLV dimensions." If you are trying to use an older FLV file that does not contain this metadata, you may notice that some of these features may not work. If this is the case, your best bet is to encode the original video to the FLV format again using the Flash 8 Video Encoder or use a utility to embed the metadata in the older FLV file.

Changing the Size of the Video

To change the layout of the video component, you can either let the component automatically size itself to the target FLV file or you can manually size the component on the Stage.

If you want the component to size itself automatically, follow the steps in the previous section for assigning the `contentPath` parameter and make sure you select the Match Source FLV Dimensions check box. Doing so will size the video component appropriately on the Stage so that you can position it visually as desired.

To size the video instance manually, follow these steps:

1. Select the video component on the Stage by clicking the Flash video icon. Open the Component Inspector (Window > Component Inspector) if it's not already open.
2. Set the `autoSize` and `maintainAspectRatio` fields to `false` in the parameter list. These parameters affect the component when the SWF file is playing. Turning off the parameters enables the video to be displayed at any size you choose.
3. With the video component still selected, choose the Free Transform tool (Q) and scale the video instance as desired.
4. Export the SWF file (Control > Test Movie) to see the results. If you want, you can play with the `maintainAspectRatio` parameter to adjust how the video appears in the custom size view.

Moving the Video to a New Location

To move the video to a new location, follow these steps:

1. Select the video component on the Stage by clicking on the Flash video icon.
2. With the Selection tool (V) you can click on the video instance and drag it to any location on the Stage. Of course, feel free to change the surrounding graphics to accommodate the change in video location.

Modifying the Buttons and Video Controls

The navigation buttons appearing along the left side of the video display are a key feature of the template. These buttons enable users to jump to predefined sections (topics) within the video. The navigation buttons label each section of the video and enable users to understand the content of the movie and how to seek to any topic desired. In this next section of the article, you'll learn how to do the following:

Understand the relationship between cue points and the navigation buttons
Add, remove, and change navigation buttons
Change FLV custom playback controls

Cue Points and Navigation Buttons

Before you jump into editing the navigation buttons, it's important to understand what is happening with the video component that makes this all work. The big picture is that you can assign sections in each FLV file that can be used for navigation and synchronization. These sections are called *cue points*. You can add them to the video when it is initially encoded to the FLV format, by using the `cuePoints` field in the FLVPlayback parameters, or by using ActionScript while the SWF file plays. For this demonstration, I will explore adding and removing cue points using the Flash Video Cue Points dialog box (see Figure 6).

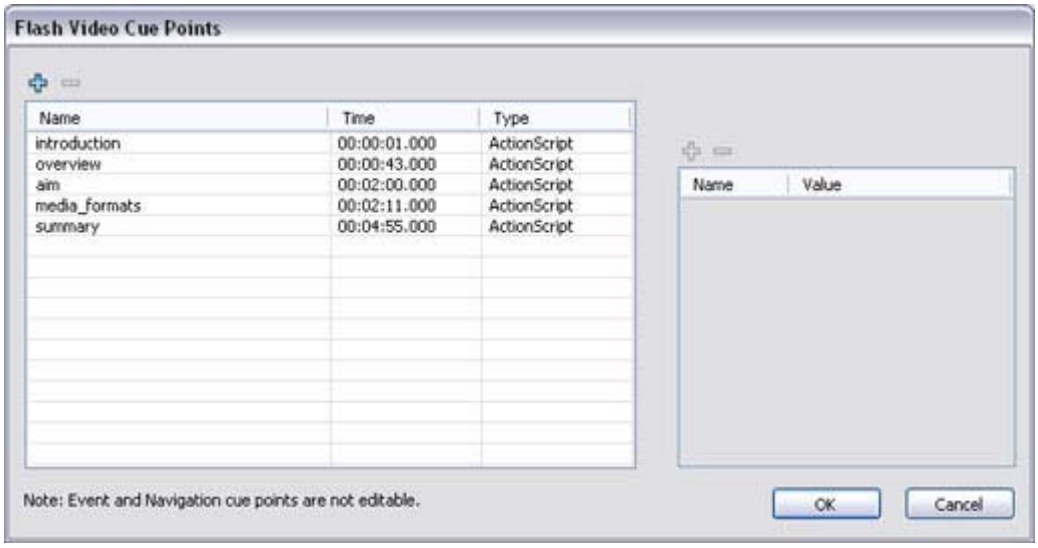

Figure 6. Flash Video Cue Points dialog box launched from the `cuePoints` field in the FLVPlayback component parameters

As you can see, named cue points can be added to each video to create sections within the video. Cue points have an associated time within the video and may also contain other data that can be referenced as cue points are encountered along the video Timeline.

To change, add, or remove cue points, follow these steps:

1. Select the video component on the Stage by clicking the Flash video icon. Open the Component Inspector (Window > Component Inspector) if it's not already open.
2. Click the `cuePoints` parameter once to select it and then click it again to launch the Flash Video Cue Points dialog box.
3. To change one of the existing fields, simply type your own cue point name and time into one of the five supplied fields. Note that there is a pattern you will use with the cue point name and the name of the navigation button that corresponds to it. If you change the names, which you most likely will, be sure to write down the new names so you can update the button instance names later.
4. To remove a cue point, select the desired field and click the minus (–) button in the dialog box. You should see the cue point disappear from the list there.
5. To add a cue point, click the plus (+) button. Notice that a new field appears at the top of the list. Enter a name and a time. The new cue point is added to the list in chronological order.

That's it. After you add your own video to the FLVPlayback component, you will adjust the cue points to match the sections of your video. You can use as many or as few cue points as you want. From there, you will update the navigation buttons to match the cue point entries.

Note: Embedding navigation cue points during encoding is more accurate than using the Flash Video Cue Points dialog box. If you are having problems with the accuracy of seeking to exact cue point times, consider encoding the video and cue points together using the Flash 8 Video Encoder instead.

Understanding the Cue Point Name and Button Instance Name Pattern

As you view the template in action, you'll notice that the down state for each button appears when the video plays past the related cue point. This template uses a naming convention to find the navigation button when a `cuePoint` event fires from the video component. The Actions layer contains an event handler function which processes `cuePoint` events.

When an event occurs, the function looks for a button whose instance name matches the cue point name plus the letters "_btn". If a button exists, it triggers the button to show its down state. For example, if a cue point event named *introduction* passes through the event handler function, then the button whose instance name is *introduction_btn* is

highlighted with its down state. There's no need to adjust any of the code. Simply create cue points and name the corresponding buttons with the cue point name plus "_btn".

Changing the Navigation Button Text

While reviewing the layers in the template, you may have noticed that two layers are set aside for the navigation buttons. The Button Text layer contains the text for each navigation button. The text is simply static text positioned and typeset in any way desired. The Button Symbols layer contains instances of the CueNameButton symbol used to create the button graphics below the text.

To change the text on a navigation button:

1. Select the Text tool (T) and click in the text block that you want to edit. You should see the text block become highlighted and show the typing cursor.
2. Edit the text as needed.
3. To change the text style, select the text block with the Selection tool and edit the text properties in the Property inspector (see Figure 7). You may also edit only a portion of text by selecting it with the Text tool and then changing the settings in the Property inspector.
4. You can continue by editing, adding, or removing text fields as needed on the Button Text layer. There is nothing functional about the labels. They should simply provide clear labels for the video.

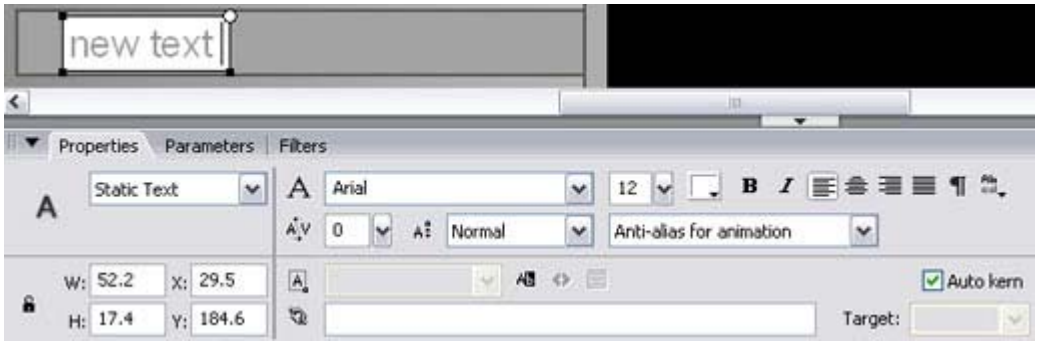

Figure 7. A selected button label during editing

Changing the Navigation Button Graphics

The CueNameButton symbol uses the new Flash 8 scaling feature called Scale 9. This enables the button to be scaled from instance to instance without seeing the effects of distortion common to most scaled movie clips. The template provides a simple button skin with an up, over, and down state that you can customize to fit your needs. You will edit your graphics in the Library and use the scale guides to control how Flash renders your graphics at different sizes (see Figure 8).

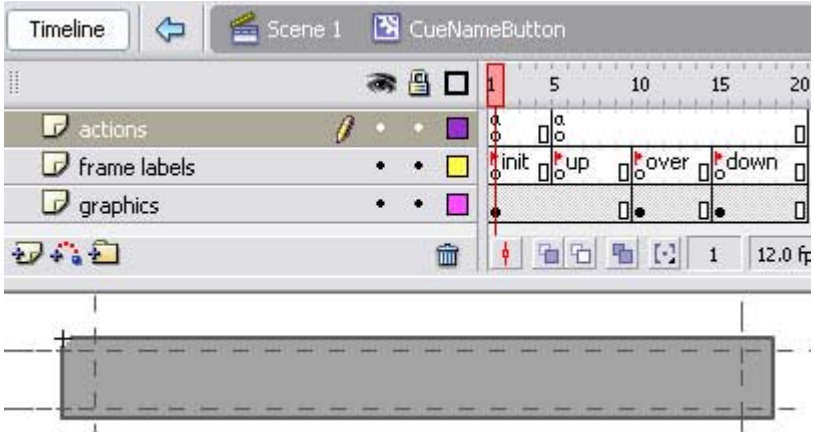

Figure 8. Navigation button Timeline; notice the guides marking off the scaling areas of the button

To edit the up, over, and down button skins, follow these steps:

1. Open the Library panel if it's not already open and double-click the CueNameButton symbol to enter its editing mode. The Timeline view should look similar to Figure 8.
2. Notice that the Timeline has four sections. Each section is marked by a frame label. The init frame loads the up graphic and some ActionScript code located on the Actions layer. The over frame contains the over graphic and the down frame contains the down graphic. During runtime the playhead jumps back and forth to the up, over, and down frames to display the correct button graphics. The up frame is staggered from the init frame so the button code on frame 1 does not reinitialize.
3. Select the up graphic on the Graphics layer at frame 1. Notice the scale guides which mark the scalable areas of the graphic. Edit the graphic as desired. You can move the guides as needed.
4. Select the over graphic on the Graphics layer at frame 10. Edit the graphic as needed. Make sure the general size of the graphics is consistent so the scale guides will work for each skin state.
5. Edit the down graphic at frame 15 in the same way.
6. When you're finished, click Scene 1 in the edit bar to return to the main Timeline.

Moving and Resizing Navigation Buttons

You can move and resize the navigation buttons any way you like. If you resize the buttons make sure that you export the SWF file to view the effects of the Scale 9 feature on your button graphics (it can be seen only in the SWF file).

Removing a Navigation Button

To remove a navigation button, simply select the button instance and the text on top of it and delete the two items from the Stage.

Adding a Navigation Button

You have two options when adding a navigation button to your file. You can either drag a new instance of the CueNameButton symbol from the Library to the Stage or copy an existing button.

To add a new navigation button and associated cue point, follow these steps:

1. Click the Buttons Symbols layer on the main Timeline of the template to select it. You will add your new button instance to this layer.
2. Open the Library panel if it's not open already and drag an instance of the CueNameButton symbol to the Stage. The symbol is located in the Video Assets/Controls folder.
3. Select the new instance on the Stage and drag it into position. Select the Free Transform tool from the tool box and resize the button so it's large enough to fit the amount of text you want to use. Take a guess at this point. You can touch up the sizing later.
4. Click the Button Text layer to select it. With the Text tool, type the button label above the button graphic you just added. Position and format the text until it looks right to you.
5. Add a new cue point to the FLVPlayback component instance. This will be the cue point that the navigation button calls when you click it. For example, add a cue point with the name **conclusion** (see the earlier section, "Cue Points and Navigation Buttons," for more information on adding cue points).
6. Click the new button instance on the Stage to select it. In the Property inspector, assign an instance name to the button using the naming convention described previously on this page (cue point name plus "_btn"). In the case of this example, the instance name should be **conclusion_btn** (see Figure 9).
7. Follow the next set of steps to complete the button setup.

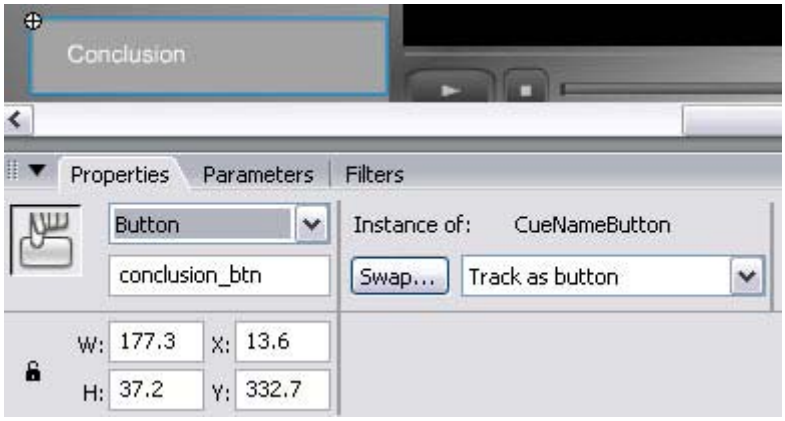

Figure 9. New button instance name in the Property inspector; note that conclusion_btn corresponds to the video cue point named conclusion

To trigger a cue point change from a new button:

1. Click the button graphic on the Stage to select it.
2. Open the Behaviors panel (Window > Behaviors). With the button still selected on the Stage, click the plus button in the Behaviors panel to add a behavior to the button. Choose the Seek to Cue Name behavior from the FLVPlayback Controls category (see Figure 10).

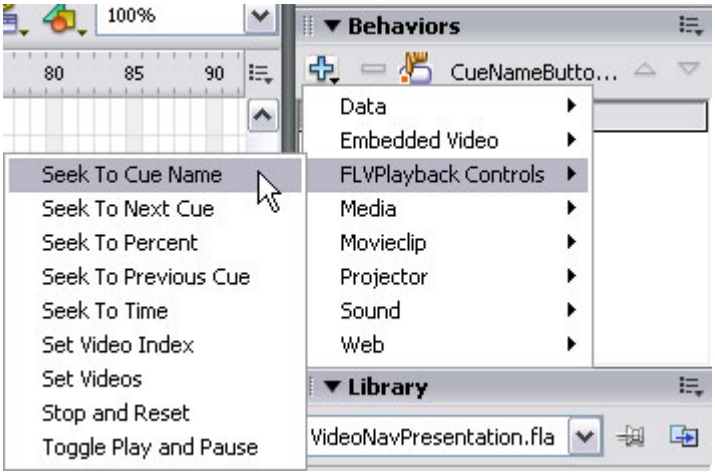

Figure 10. The nine FLVPlayback behaviors created for these templates

3. In the dialog box that appears, enter the name of the cue point that a button click is supposed to trigger. In the case of the example above, the name would be **conclusion**. Before you exit the dialog box, select the FLVPlayback instance name display (see Figure 11).

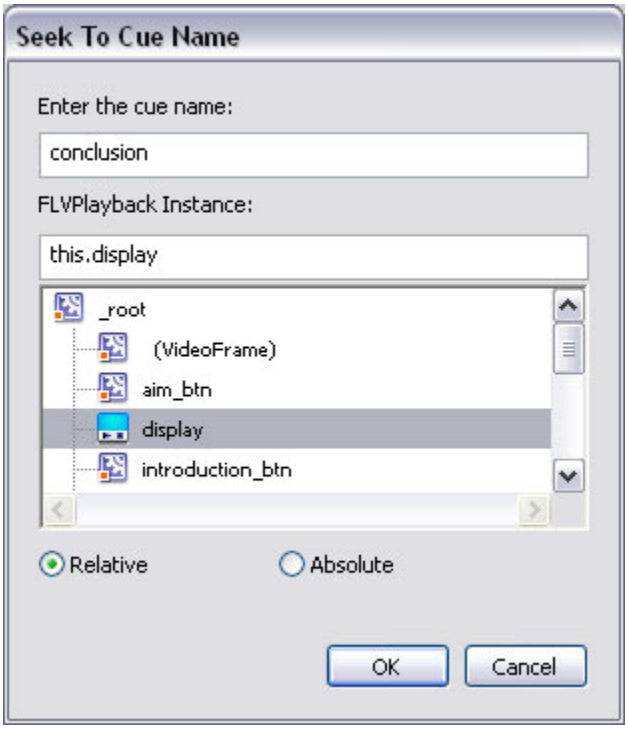

Figure 11. Seek to Cue Name dialog box used to enter parameters for the behavior

4. Click OK to exit the dialog box.

That's it. Your new navigation button should trigger the video to jump to the corresponding cue point.

Note: The FLVPlayback behaviors need to be installed by running the FLVPlaybackBehav.mxp file. After you install the behaviors, they should appear in the Behaviors panel shown in Figure 10.

Adding and Removing Custom User Interface Controls

The template contains a basic set of video controls, including a play/pause button, stop button, seek bar, volume bar, and mute button. These controls are placed on the FLV Controls layer alongside the FLVPlayback instance. The controls are associated with the FLVPlayback instance using a small amount of ActionScript code on the Actions layer. You can remove any or all of the controls without making other adjustments to the template. If you want to add additional custom user interface components, you will need to name their instances and update the code on the Actions layer to associate them with the video component.

To remove all the user interface controls, follow these steps:

1. Select all the controls on the FLV Controls layer except for the FLVPlayback component.
2. Press the Delete button to remove the controls.
3. Although it's not mandatory, it would be good form to clean up the code on the Actions layer. Select the keyframe on the first frame of the Actions layer and open the Actions panel (Window > Actions).
4. In the Actions panel, select and delete the following code:

```
//-----------------
// Video control assignment
//-----------------
// Use the skin assignment properties of the FLVPlayback
// instance to associate the controls with the player.
display.playPauseButton = play_btn;
display.stopButton = stop_btn;
display.seekBar = seek_bar;
display.volumeBar = volume_bar;
display.muteButton = mute_btn;
```

You may add any combination of FLVPlayback custom user interface controls by dragging instances from the

Components panel to the Stage. You must name the instance and assign the instance name to the FLVPlayback instance using assignment properties as seen in the code above.

To add a buffering bar, you would follow these steps:

1. Drag a BufferingBar component from the Components panel to the Stage.
2. Select the new instance and move it into position.
3. With the instance still selected, name it in the Property inspector (for example, **bufferingBar_mc**).
4. The last step is to assign the instance name to the FLVPlayback instance named display using a bit of ActionScript. Select the first frame of the Actions layer and return to the Actions panel. At the top of the script, add the following line of code:

```
// Assign the buffering bar to the video component
display.bufferingBar = bufferingBar_mc;
```

The FLVPlayback component has a handful of control assignment properties that you can use in a similar way as above to assign the other control types to the video component. The property list includes the following:

1. `playButton`
2. `pauseButton`
3. `playPauseButton`
4. `stopButton`
5. `muteButton`
6. `backButton`
7. `forwardButton`
8. `volumeBar`
9. `seekBar`
10. `bufferingBar`

You can customize the FLVPlayback custom controls easily to match whatever look and feel you need. For this template I changed the green highlight color to a red to match the video theme better. See my article, Customizing the FLVPlayback Component, for more information on skinning options.

Customizing the Background Graphics

One of the easiest changes to make to the template is modifications to the look and feel of the background graphics. Changing the color of the chrome (gradient) graphics in the background can really go a long way in creating a unique-looking template that fits your company's branding and tone. Simple changes to the template may include:

Changing the graphic which frames the video and video controls
Changing the background graphics and logo

Changing the Video Background Graphic

As you've seen in previous sections of this article, the playback component and the video controls are separate. This makes customizations easy by enabling you to move the controls to any location, swap different types of controls in and out of the template, and create any background graphic you like. For the purposes of this template, I created a simple custom background inside the VideoFrame symbol in the Library (see Figure 12).

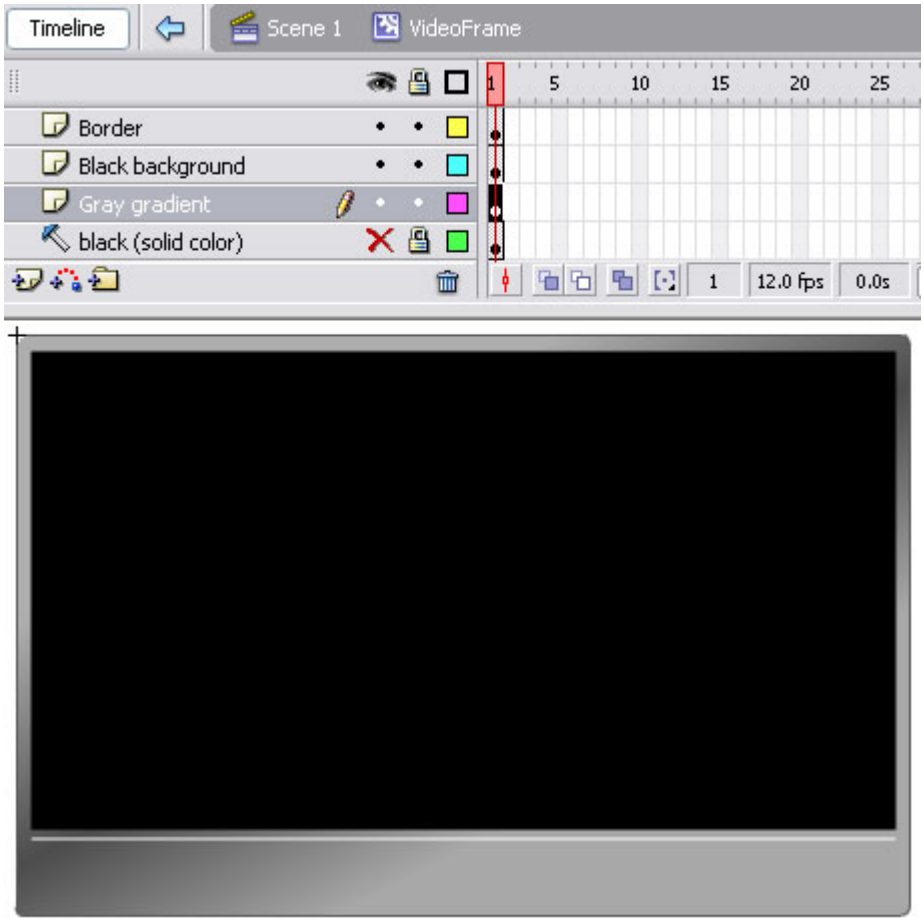

Figure 12. VideoFrame symbol editing area in the Library

To edit the video background graphic, follow these steps:

1. Hide the video components that overlay the background graphic. You can do this by hiding the FLV Controls layer on the main Timeline.
2. Double-click the video background graphic to enter its editing mode in the Library. You should see a similar view to the image in Figure 12. The edit bar shows that you are now editing the VideoFrame symbol. You can return to the main Timeline by clicking the Scene 1 button in the edit bar.
3. Take a quick look at the layers provided. The Black (Solid Color) layer is currently turned off from exporting using a guide. This layer holds the basic shape I used for the background. The Gray Gradient layer shows the same shape with a chrome gradient applied to it for effect. You can edit the gradient using the Color Mixer panel if you like. The Black Background layer displays a black rectangle that's the same size as the video. It creates the illusion that the video is in place before it actually loads to the screen. Finally, the Border layer applies a thin stroke around the gradient shape.
4. Feel free to add and remove layers as needed. Use the existing graphics or add your own. When you're finished, click the Scene 1 button on the edit bar at the top of the Timeline to return to the primary editing area for the templates.

Changing the Background and Logo

This is the easy part. Edit the layers contained in the Background Graphics Layer folder (see Figure 13). You can add and remove layers and graphics as needed.

Figure 13. Background graphics with the video controls hidden for clarity

To change the background and logos, follow these steps:

1. To make things easier on yourself, hide the video controls by hiding the Controls folder. To do this, click the dot below the Eyeball icon on the Controls folder. The resulting view of the Stage should look like Figure 14. You may want to collapse the Controls folder to get those layers out of your way.

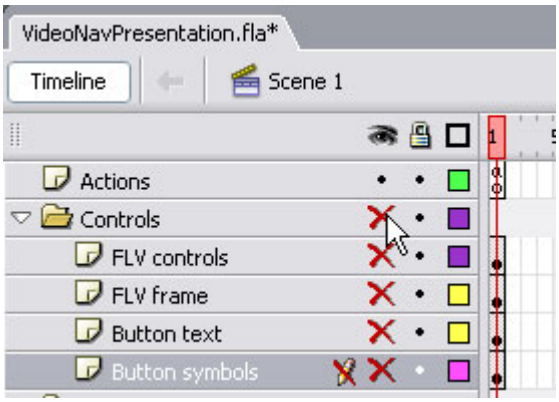

Figure 14. Hiding the video control layers while editing the background graphics

2. Click the Logo layer and delete the existing logo. Add your own logo graphics as desired. You can import new graphics by selecting File > Import > Import to Stage from the menu.
3. You may want to delete the Macromedia layer or update the copyright information to suit your needs.
4. Edit the graphics on the Gray Top and Gradient Back layers. Feel free to change the colors and gradients or delete them entirely and add new graphics that match your project.

Where to Go from Here

Now that you have made the presentation your own, there are many ways to craft it to target your needs more specifically:

Animate the text and graphics on the screen in unique ways to complement the video
Add multiple videos and allow users to select which video to watch
Add and remove custom video controls using the FLVPlayback custom user interface components
Add extended functionality using the FLVPlayback ActionScript API or the FLVPlayback behaviors (see my article, Controlling Flash Video with the FLVPlayback Behaviors)
Change the look and feel of the video controls using the FLVPlayback skinning features (see my article,

Customizing the FLVPlayback Component)

Spend some time with the tutorial Flash Video Template: Showcase Website for Personal Video for even more ideas on fun, effective ways to communicate your message better. Make sure you also explore the other Flash video templates provided in the Flash Developer Center.

About the author

Dan Carr is owner, lead developer, and trainer for Dan Carr Design in San Francisco. With years of history developing for Macromedia, Dan has created a range of features for Flash, including e-learning templates and UI components for Flash MX Professional 2004. Dan teaches Flash design and ActionScript classes in San Francisco and develops e-learning and web applications for the public, as well as for Macromedia product teams.