



## Creating Animation and Graphics using Macromedia Flash: Intermediate Level WORKSHOP

Use the Flash animation program to create vector-based graphics, animation, and interactive movies for the Web.

- ✓ Creating and Editing Symbols
- ✓ Creating and Testing Buttons
- ✓ Interactive Movies
- ✓ Animation

**Student Technology Services** provides AASU students a wide range of technology resources, services, and information, so students are able to incorporate technology into their learning environment and course work.

**Student Help Desk** provides in-depth phone and email technical support and resources to students regarding WebCT, Microsoft Office programs, and student email.

**Computing Labs** are provided in MCC Annex and UH 112 for students to use to complete academic course work.

**Orientations/Workshops** are now offered regarding WebCT, computer and Internet basics, and Microsoft Office programs. Students are encouraged to visit [www.sts.armstrong.edu](http://www.sts.armstrong.edu) to view an updated schedule of orientations and workshops.

### **CONTACT STUDENT TECHNOLOGY SERVICES:**

[www.sts.armstrong.edu](http://www.sts.armstrong.edu)

**STUDENT HELP DESK**

912.927.5321

Toll-free 1.866.533.6353

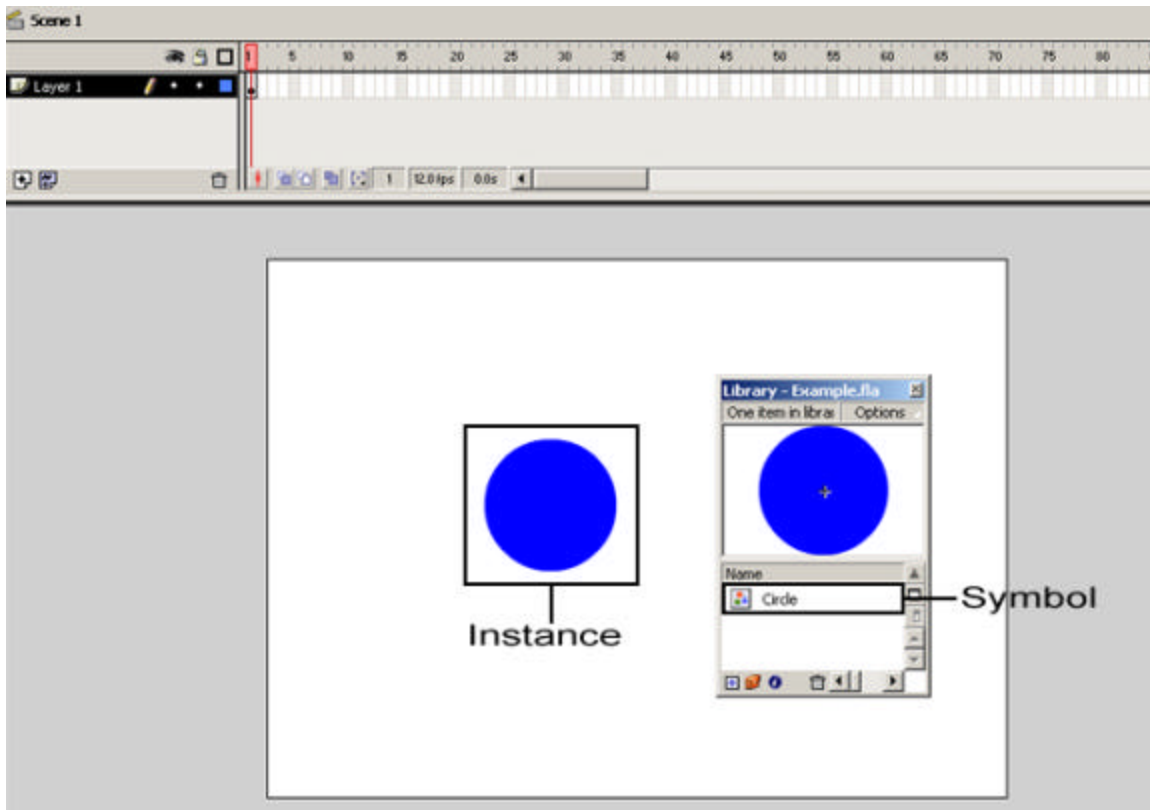
[techhelp@mail.armstrong.edu](mailto:techhelp@mail.armstrong.edu)

# Creating Animations and Graphics using Macromedia Flash: Intermediate Level

## Symbols

### Overview of Symbols and Instances

In Flash, a *Symbol* is a graphic, button or movie clip that is created once and can be reused throughout a movie or in other movies. Any *Symbol* created automatically becomes part of the *Library*. An *Instance* is a copy of a *Symbol* located on the stage, as depicted in *Figure 1*, or nested inside another *Symbol*. An *Instance* can be very different from its *Symbol* in color, size, and function. Editing a *Symbol* updates all of its *Instances*, but editing an *Instance* of a *Symbol* only updates that *Instance*.






**Figure 1. Instance vs. Symbol**

### Symbol Behaviors

Each symbol has its own timeline, stage, and layers. When you create a symbol, you determine its behavior according to how you intend to use it in a movie. Available symbol behaviors are described in *Table 1*.

**Table 1. Symbol Behaviors**

Behavior	Description
	Movie clip symbols are used to create reusable pieces of animation. Movie clips have their own multiframe Timeline that plays independent of the main movie's Timeline – think of them as mini-movies inside a main movie that can contain interactive controls, sounds, and even other movie clip instances. You can also place movie clip instances inside the Timeline of a button symbol to create animated buttons.
	Button symbols are used to create interactive buttons in the movie that respond to mouse clicks, rollovers, or other actions.
	Graphic symbols are used for static images and to create reusable pieces of animation that are tied to the Timeline of the main movie. They operate in sync with the Timeline of the main movie. Interactive controls and sounds will not work in a graphic symbol's animation sequence.

### Creating Symbols

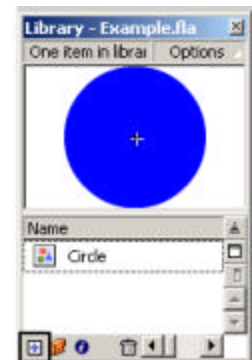
A symbol can be created from objects that already exist on the stage. To create a symbol with selected objects:

1. Select an object or several objects on the stage, and do one of the following:
  - Select **Insert** → **Convert to Symbol**.
  - Press <F8>.
2. In the *Symbol Properties* dialog box, type the name of the symbol, and select the behavior – *Movie Clip*, *Button*, or *Graphic*.
3. Click **OK**.

Flash adds the symbol to the *Library*, and the selection on the stage becomes an instance of the symbol. The object can no longer be edited on the stage. Instead, it must be opened in symbol-edit mode.

A symbol is not always created from objects that exist on the stage. To create a blank symbol:

1. Do one of the following:
  - Select **Insert** → **New Symbol**.
  - Click on the *New Symbol* button located on the *Library* window (Figure 2).
  - Select **New Symbol** from the *Library Options* menu.
  - Press <Ctrl> + <F8>
2. In the *Symbol Properties* dialog box, type the name of the symbol, and select the behavior – *Movie Clip*, *Button*, or *Graphic*.
3. Click **OK**.



New Symbol button  
**Figure 2. Library Window**

- Flash adds the symbol to the *Library* and switches to symbol-edit mode. In symbol-edit mode, the name of the symbol appears above the top left corner of the window, above the Timeline, and a cross hare indicates the symbol's registration point.
- To create the symbol content, use the Timeline, draw with the drawing tools, import media, or create instances of other symbols.
- After editing the symbol content, do one of the following to return to movie-editing mode:
  - Select **Edit → Edit Movie**.
  - Click on the *Scene* button in the upper left corner of the document window.

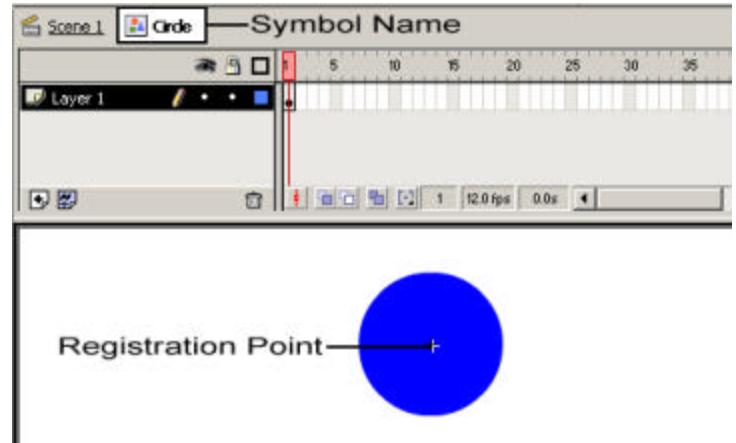
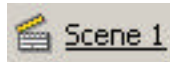
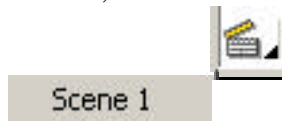


Figure 3. Symbol-edit mode



- Click on the *Edit Scene* button in the upper right corner of the document window, and select the scene from the menu.



### Adding an Instance

To add an *Instance* of a *Symbol* to a movie, do the following:

- Select the layer in the timeline. **Note: Instances can only be placed in keyframes on the current layer. If you do not select a keyframe, the instance will be added to the first keyframe to the left of the current frame.**
- Open the *Library* by doing one of the following:
  - Select **Window → Library**.
  - Press <Ctrl> + <L>.
- Drag the *Symbol* from the *Library* onto the stage.

After creating an instance of a symbol, use the *Instance* panel (**Windows → Panels → Instance**) to specify color effects, assign actions, set the graphic display mode, or change the behavior of the instance. The behavior of the instance is the same as the symbol behavior, unless you specify otherwise. Any changes you made affect only the instance and not the symbol.


### Editing Symbols

When a symbol is edited, all instances of that symbol are updated. Editing a symbol is done in one of two ways: *Edit in Place* and *Edit in New Window*. When editing a

symbol in place, the symbol is edited in context with other objects on the stage. To edit a symbol in place, double-click on the symbol on the stage. Other objects are dimmed to distinguish them from the symbol being edited.

When editing a symbol in a new window, Flash switches to symbol-edit mode. To edit a symbol in a new window:

1. Do one of the following:
  - Right click on the symbol, and select **Edit in New Window** from the context menu.
  - Double click on the symbol's icon in the *Library* window.
  - Select an instance of the symbol on the stage, and click the *Edit Symbol* button at the bottom of the *Instance* panel.
  - Select an instance of the symbol on the stage, and select **Edit → Edit Symbol**.
2. The symbol opens in symbol-edit mode.
3. Edit the symbol contents.
4. To return to movie-editing mode, do one of the following:

- Select **Edit → Movie**.
  - Click on the *Scene* button in the upper left corner of the document window.
- 
- Click on the *Edit Scene* button in the upper right corner of the document window, and select the scene from the menu.



## Buttons

### Overview of Buttons

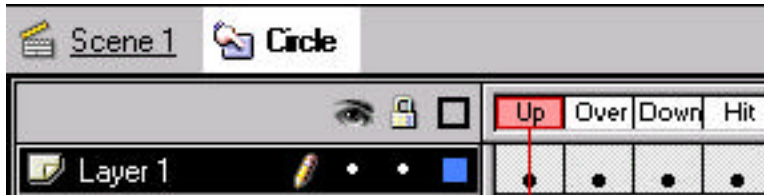
Buttons are four-frame, interactive movie clips. When the button behavior is selected for a symbol, Flash creates a Timeline with four frames. The first three frames display the button's three possible states. The fourth frame defines the active area of the button. The timeline does not play. However, it reacts to pointer movement and actions by jumping to the appropriate frame.

To make a button interactive in a movie, an instance of the button symbol must be placed on the stage and actions assigned to the instance. The actions must be assigned to the instance of the button in the movie, not to the frames in the button's Timeline.

### Creating a Button

Each frame in the timeline of a button has a specific function:

- The first frame is the *Up* state, representing the button whenever the pointer is not over the button.
- The second frame is the *Over* state, representing the button's appearance whenever the pointer is over it.
- The third frame is the *Down* state, representing the button's appearance as it is clicked.
- The fourth frame is the *Hit* state, which defines the area that will respond to the mouse click. This area is invisible in the movie.



**Figure 4. Button states**

To create a button:

1. Select **Edit** → **Deselect All** to ensure that nothing is selected on the stage.
2. Select **Insert** → **New Symbol**, or press <Ctrl> + <F8>.
3. In the *Symbol Properties* dialog box, enter the name for the new button symbol, and for *Behavior*, select *Button*. Flash switches to symbol-edit mode. The timeline header changes to display four consecutive frames labeled *Up*, *Over*, *Down*, and *Hit*. The first frame, *Up*, is a blank keyframe.
4. To create the *Up* state button image, use the drawing tools, import a graphic, or place an instance of another symbol on the Stage. **Note: You can use a graphic or movie clip symbol in a button, but you cannot use another button in a button. Use a movie clip symbol if you want the button to be animated.**
5. Click the second frame, labeled *Over*, and select **Insert** → **Keyframe**. Flash inserts a keyframe that duplicates the contents of the *Up* frame.
6. Change the button image for the *Over* state.
7. Repeat steps 5 and 6 for the *Down* frame and the *Hit* frame. The *Hit* frame is not visible on the Stage, but it defines the area of the button that responds when clicked. Make sure that the graphic for the *Hit* frame is a solid area large enough to encompass all the graphic elements of the *Up*, *Down*, and *Over* frames. It can also be larger than the visible button.
8. When finished, select **Edit** → **Edit Movie**. Drag the button symbol out of the *Library* window to create an instance of it in the movie.

### **Enabling, Editing, and Testing Buttons**

To make buttons easier to select and work with, Flash keeps them disabled by default. When a button is disabled, clicking the button selects it. When a button is enabled, it responds to the mouse events that have been specified as if the movie were playing.

Though it is possible to select enabled buttons, in general, it is best to disable buttons while working, and enable buttons to quickly test their behavior.

To enable and disable buttons, select **Control → Enable Simple Buttons**. A check mark appears next to the command to indicate buttons are enabled. (Select the command again to disable buttons.) Any buttons on the stage now respond. As the pointer is moved over the button and the button's active area is clicked, Flash displays the *Over* frame and the *Down* frame, respectively.

To move or edit the enabled button:

1. Select the button by using the *Arrow* tool to draw a selection rectangle around the button.
2. Do one of the following:
  - Use the arrow keys to move the button.
  - Select **Window → Panels → Instance** to edit the button, or press <Alt> and double click on the button.

To test a button, do one of the following:

- Select **Control → Enable Simple Buttons**. Move the pointer over the enabled button to test it.
- Select the button in the *Library* window and click the Play button in the *Library* preview window.
- Select **Control → Test Scene** or **Control → Test Movie**.

## Adding Sound

### Overview

Flash offers a number of ways to use sounds. Sounds can play continuously, independent of the Timeline, or sound tracks can be synchronized to animation. Sounds can also be attached to buttons to make them more interactive. Additionally, sounds can be made to fade in and out for a more polished sound track.

There are two types of sounds in Flash: event sounds and stream sounds. An event sound must download completely before it begins playing, and it continues playing until explicitly stopped. Stream sounds begin playing as soon as enough data for the first few frames has been downloaded; stream sounds are synchronized to the Time line for playing on a Web site.

### Importing Sounds

To import a sound into a Flash movie, do the following:

1. Select **File → Import**.

2. In the *Import* dialog, browse to locate the file, select the file, and click **Open**. The sound file is added to the *Library*.

### **Adding Sounds to Buttons**

Sounds can be associated with the different states of a button symbol. Because sounds are stored with the symbol, they work for all instances of the symbol.

To add a sound to a button:

1. Double-click on the button symbol in the *Library* to open the button in symbol-edit mode.
2. In the button's Timeline, add a layer for sound.
3. In the sound layer, create keyframe to correspond to the button state to which you want to add the sound. For example, to play a sound when a button is clicked, add a keyframe in the frame labeled *Down*.
4. Click the keyframe just created.
5. Open the *Sound* panel (**Window** → **Panels** → **Sound**).
6. In the *Sound* drop down, select the desired sound file.
7. In the *Sync* drop down, select *Event*.

## **Interactive Movies**

### **Overview**

In simple animation, Flash plays the scenes and frames of a movie sequentially. In an interactive movie, the audience uses the keyboard, the mouse, or both to jump to different parts of a movie, movie objects, enter information in forms, and perform many other interactive operations. Sets of instructions that run when a specific event occurs create interactive movies. The events that can trigger an action are either the playhead reaching a frame, or the user clicking a button or pressing keys on the keyboard.

Actions for a button, a movie clip, or a frame are set up in the *Actions* panel. The Actions panel controls can be used in two modes: *Normal* and *Expert*. Instructions can be in the form of a single action, such as instructing a movie to stop playing, or a series of actions, such as first evaluating a condition and then performing an action. Many actions require little programming knowledge or experience to set up. Other actions require some familiarity with programming languages and are intended for advanced development.

### **About ActionScript**

Flash uses a scripting language, called ActionScript, to add interactivity to a movie. Similar to JavaScript, ActionScript is an object-oriented programming language. In object-oriented scripting, information is arranged into groups called classes. You can create multiple instances of a class, called objects, to use in your scripts. You can use ActionScript's predefined classes and create your own.

When creating a class, you define all of the properties (characteristics) and methods (behaviors) of each object it creates, just as real-world objects are define. For example, a person has properties such as gender, height, and hair color and methods such as talk, walk, and throw. In this example, “person” is a class and each individual person is an object or instance of that class.

### **Using the Actions Panel**

The *Actions* panel is utilized to create and edit actions for an object or frame using two different editing modes: *Normal* and *Expert*. In *Normal* mode, you can write actions using parameter (argument) fields that prompt you for the correct arguments. In *Expert* mode, you can write and edit actions directly in the text box, much like writing code with a text editor.

To display the *Actions panel*, select **Window → Actions**. Selecting a frame, button, or movie clip instances makes the *Actions* panel active. The *Actions* panel title changes to *Object Actions* if a button or movie clip is selected, and to *Frame Actions* if a frame is selected.

To select an actions editing mode:

1. With the *Actions* panel displayed, click the arrow in the upper right corner of the panel to display the pop-up menu.
2. Select *Normal* mode or *Expert* mode from the popup menu.

Each script maintains its own mode. For example, if you script one instance of a button in *Normal* mode and another in *Expert* mode, switching between the selected buttons will switch the panel’s mode.

### **Assigning Actions to Objects**

You can assign an action to a button or a movie clip to make an action execute when the user clicks a button or rolls the pointer over it, or when the movie clip loads or reaches a certain frame. The action is assigned to an instance of the button or movie clip, and other instances of the symbol aren't affected.

To use the *Actions* panel to assign an action to a button or movie clip:

1. Select the button or movie clip instance, and select **Window → Actions**. If the selection is not a button instance, a movie clip instance, or frame, or if the selection includes multiple objects, the *Actions* panel will be dimmed.
2. In the *Toolbox* list on the left side of the panel, click the *Basic Actions* category to display the basic actions.
3. To assign an action, do one of the following:
  - Double click an action in the *Basic Actions* category.

- Drag an action from the *Basic Actions* category on the left side to the *Actions* list on the right side of the panel.
- Click the Add (+) button and select an action from the popup menu.

### **Basic Actions for Navigation and Interaction**

The basic actions in the *Actions* panel help control navigation and user interaction in a movie by selecting actions and having Flash write the ActionScript. The basic actions include:

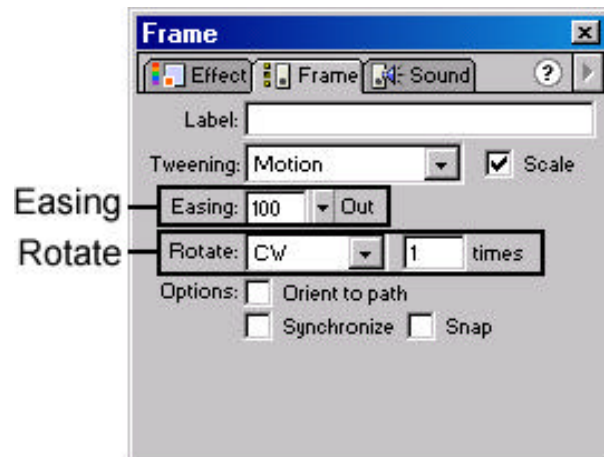
- The *Go To* action, which jumps to a frame or scene.
- The *Play* and *Stop* actions, which play and stop movies
- The *Get URL* action, which jumps to a different URL.
- The *If Frame Is Loaded* action, which checks whether a frame is loaded.
- The *On Mouse Event* action, which assigns a mouse event or keyboard key triggering an action.

## **Animation**

### **Motion Tween Properties**

Available properties for *Motion Tweening* include *Rotate* and *Easing*. Rotation can be applied to a *Motion Tween* so that an object is rotated in a clockwise or counter clockwise fashion. After inserting a *Motion Tween*, open the *Frame* panel (**Window → Panels → Frame**), select the direction from the *Rotate* drop down list, and type in the number of times you want the object to rotate.

The *Easing* property allows you to specify the rate of motion from start to finish. With a value of -100, objects accelerate towards the finish line while the reverse happens if the value is set to 100.



### **Fading In and Out**

In Flash, creating a Motion Tween and changing the alpha channel in either the initial or final keyframe of an instance accomplishes the effect of fading in or out. To create the effect of fading in:

1. Select and click inside of the layer and initial frame. (You may need to create a new layer and keyframe.)

2. Do one of the following:
  - If the desired symbol already exists, drag the symbol from the *Library* onto the stage.
  - Create a new symbol from the desired objects on the stage.
3. On the same layer and several frames later, create a new keyframe.
4. Click inside of the initial keyframe.
5. Open the *Frame* panel (**Window** → **Panels** → **Frame**), and for *Tweening*, select *Motion*.
6. Open the *Effect* panel (**Window** → **Panels** → **Effect**), or from the *Frame* panel, click on the *Effect* tab.
7. From the drop down list, select *Alpha*.
8. In the next drop down list, either type in a percent value or drag the bar to the desired percentage. (Note: A setting of 0% makes the object totally invisible.)
9. Move the play head back and forth to preview the animation.

To create the effect of fading out, keep the alpha channel setting of the object for the initial keyframe to 100%, and change the alpha channel setting for the final keyframe to less than 100%.

## **Publishing Flash Movies for the Web**

An HTML document is required to play a Flash movie in a Web browser and specify browser settings. If you are experienced with HTML, you can change or enter HTML parameters manually in an HTML editor, or create your own HTML files to control a Flash movie.

### **OBJECT and EMBED Tags**

To display a Flash Player movie in a Web browser, an HTML document must use the OBJECT and EMBED tags with proper parameters. The OBJECT tag is used by Internet Explorer (Windows), and the EMBED tag is used with Netscape Navigator (Windows and Mac) and Internet Explorer (Mac).

Hand coding HTML pages with OBJECT and EMBED tags is not necessary when using Macromedia Flash 4 or later. The Publish command (**File** → **Publish**) will automatically create an HTML document that contains the required HTML tags for browsers to access a Flash movie on the web.

Macromedia Dreamweaver can easily embed a SWF movie into an HTML document. Dreamweaver also provides more control over layout and design of the page that contains the Macromedia Flash movie that the Publish command in Flash.

For OBJECT, four settings (HEIGHT, WIDTH, CLASSID, and CODEBASE) are attributes that appear within the OBJECT tag; all others are parameters that appear in separate, named PARAM tags. For example:

```
<OBJECT CLASSID="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" WIDTH=550 HEIGHT=400
CODEBASE="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0">
<PARAM NAME="MOVIE" VALUE="Shape Tween.swf">
<PARAM NAME="QUALITY" VALUE="HIGH">
<PARAM NAME="BGCOLOR" VALUE="#66CCFF">
</OBJECT>
```

For the EMBED tag, all settings (such as HEIGHT, WIDTH, QUALITY, and LOOP) are attributes that appear between the angle brackets of the opening EMBED tag. For example:

```
<EMBED SRC="Shape Tween.swf" WIDTH="550" HEIGHT="400"
LOOP="true" QUALITY="high" BGCOLOR="#66CCFF"
TYPE="application/x-shockwave-flash"
PLUGINSPPAGE="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=ShockwaveFlash"></EMBED>
```

To use both tags together (recommended), position the EMBED tag just before the closing OBJECT tag, as follows:

```
<OBJECT CLASSID="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" WIDTH=550 HEIGHT=400
CODEBASE="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0">
<PARAM NAME="MOVIE" VALUE="Shape Tween.swf">
<PARAM NAME="QUALITY" VALUE="HIGH">
<PARAM NAME="BGCOLOR" VALUE="#66CCFF">

<EMBED SRC="Shape Tween.swf" WIDTH="550" HEIGHT="400"
LOOP="true" QUALITY="high" BGCOLOR="#66CCFF"
TYPE="application/x-shockwave-flash"
PLUGINSPPAGE="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=ShockwaveFlash"></EMBED>

</OBJECT>
```

If you use both the OBJECT and EMBED tags, use identical values for each attribute or parameter to ensure consistent playback across browsers. The parameter `swflash.cab#version=5,0,0,0` is optional, and you can omit it if you don't want to check for version number.

Tag attributes and parameters describe the HTML created by the Publish command. You can refer to *Table 2* as you write your own HTML to insert in Flash moves. Unless noted, all items apply to both OBJECT and EMBED tags. Optional entries are noted.

**Table 2. Tag attributes and parameters**

Tag	Parameter(s)	Description
SRC	movie.swf	Specifies the name of the movie to be loaded. EMBED only.
MOVIE	movie.swf	Specifies the name of the movie to be loaded. OBJECTS only.
CLASSID	clsid:#	Identifies the ActiveX control for the browser. The value must be entered exactly as shown. clsid:D27CDB6E-AE6D-11cf-96B8-444553540000 OBJECTS only.
WIDTH	# or %	Specifies the width of the movie either in pixels or percentage of the browser window. Because Flash movies are scalable, their quality won't degrade at different sizes if the aspect ratio is maintained.
HEIGHT	# or %	Specifies the height of the movie either in pixels or percentage of the browser window. Because Flash movies are scalable, their quality won't degrade at different sizes if the aspect ratio is maintained.
CODEBASE	URL	Identifies the location of the Flash Player ActiveX control so that the browser can automatically download it if it is not already installed. The value must be entered exactly as shown: <a href="http://active.macromedia.com/flash5/cabs/swflash.cab#version=5,0,0,0">http://active.macromedia.com/flash5/cabs/swflash.cab#version=5,0,0,0</a> OBJECT only.
PLUGINSPAGE	URL	Identifies the location of the Flash Player plug-in so that the user can download it if it is not already installed. The value must be entered exactly as shown: <a href="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=ShockwaveFlash">http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=ShockwaveFlash</a> EMBED only
SWLIVECONNECT	true   false	(Optional) Specifies whether the browser should start Java when loading the Flash Player for the first time. The default value is false if this attribute is omitted. If you use JavaScript and Flash on the same page, Java must be running for the FSCommand to work. However, if you are using JavaScript only for browser detection or another purpose unrelated to FSCommand actions, you can prevent Java from starting by setting SWLIVECONNECT to false. You can also force Java to start when you are not using JavaScript with Flash by explicitly setting SWLIVECONNECT to true. Starting Java substantially increases the time it takes to start a movie; set this tag to true only when necessary. EMBED only.

PLAY	true   false	(Optional) Specifies whether the movie begins playing immediately on loading in the browser. If your Flash movie is interactive, you may want to let the user initiate the play by clicking a button or performing some other task. In this case, set the Play attribute to false to prevent the movie from starting automatically. The default value is true if this attribute is omitted.
LOOP	true   false	(Optional) Specifies whether the movie repeats indefinitely or stops when it reaches the last frame. The default value is true if this attribute is omitted.
QUALITY	low   high   autolow   autohigh   best	(Optional) Specifies the level of anti-aliasing to be used during playback of your movie. Because anti-aliasing requires a faster processor to smooth each frame of the movie before it is rendered on the viewer's screen, choose a value based on whether speed or appearance is your top priority. <ul style="list-style-type: none"> <li>• Low favors playback speed over appearance and never uses anti-aliasing.</li> <li>• Autolow emphasizes speed first, but improves appearance whenever possible.</li> <li>• Autohigh emphasized playback speed and appearance equally at first but sacrifices appearance for playback speed if necessary.</li> <li>• High favors appearance over playback speed and always applies anti-aliasing.</li> <li>• Best provides the best play quality and does not consider playback speed.</li> </ul> The default value for Quality is high if this value is eliminated.
BGCOLOR	#RRGGBB (Hex. RGB number)	(Optional) Specifies the background color of the movie. Use this attribute to override the background color setting specified in the Flash file. This attribute does not affect the background color of the HTML page.
SCALE	showall   noborder   exactfit	(Optional) Defines how the movie is placed within the browser window when WIDTH and HEIGHT values are percentages. <ul style="list-style-type: none"> <li>• Default (Show all) makes the entire movie visible in the specified area without distortion, while maintaining the original aspect ratio of the movie. Borders may appear on two sides of the movie.</li> <li>• No Border scales the movie to fill the specified area, without distortion but possibly with some cropping, while maintaining the original aspect ratio of the movie.</li> <li>• Exact Fit makes the entire movie visible in the specified area without trying to preserve the</li> </ul>

		original aspect ratio. Distortion may occur.
ALIGN	L   R   T   B	<p>Specifies the ALIGN attribute for the OBJECT, EMBED, and IMG tags and determines how the Flash movie is positioned in the browser window.</p> <ul style="list-style-type: none"> <li>• Default centers the movie in the browser window and crops edges if the browser window is smaller than the movie.</li> <li>• Left, Right, Top, and Bottom align the movie along the corresponding edge of the browser window and crop the remaining three sides as needed.</li> </ul>
SALIGN	L   R   T   B   TL   TR   BL   BR	<p>(Optional) Specifies where a scaled Flash movie is positioned within the area defined by the WIDTH and HEIGHT settings.</p> <ul style="list-style-type: none"> <li>• L, R, T, and B align the movie along the left, right, top, or bottom edge, respectively, of the browser window and crop the remaining three sides as needed.</li> <li>• TL and TR align the movie to the top left and top right corner, respectively, of the browser window and crop the bottom and remaining right and left side as needed.</li> <li>• BL and BR align the movie to the bottom left and bottom right corner, respectively, of the browser window and crop the top and remaining right or left side as needed.</li> </ul>
BASE	Base directory or URL	<p>(Optional) Specifies the base directory or URL used to resolve all relative path statements in the Flash movie. This attribute is helpful when your Flash Player movies are kept in a different directory from your other files.</p>
MENU	true   false	<p>(Optional) Specifies the menu type that is displayed when the viewer right clicks the movie area on the browser:</p> <ul style="list-style-type: none"> <li>• True displays the full menu, allowing the user a variety of options to enhance or control playback.</li> <li>• False displays a menu that only contains the about Flash option</li> </ul>
WMODE	Window   Opaque   Transparent	<p>(Optional) Takes advantage of the transparent movie, absolute positioning, and layering capabilities of recent browsers:</p> <ul style="list-style-type: none"> <li>• Window plays the movie in its own rectangular window on a Web Page.</li> <li>• Opaque makes the movie hide everything behind it on the page.</li> <li>• Transparent makes the background of the HTML page show through all the transparent portions of</li> </ul>

		the movie, and may slow animation performance.
--	--	--

## Exercise 1. Animated Buttons

Making an animated button is similar to making a simple button. However, instead of drawing objects or using graphic symbol instances, Movie Clip symbols are utilized to create the animation.

A movie symbol is a self-contained movie whose timeline moves independently of the main movie's timeline. Movie Clip symbols can contain animations, and here, we look at creating two simple movie symbols. The first symbol, which we will call *fade-in*,

consists of an animated rectangular graphic, which is semitransparent at the start and then becomes clear. The second, called *fade-out*, contains the same rectangular graphic, which fades out.

The first step is to make the rectangular graphic symbol. Open a new file, and in the first frame, draw a rectangle with no outline and a blue fill color. Select the *Text Tool*, change the *Fill Color* to white, and type in the text *BUTTON*. Align both the rectangle and the text to the center of the page. Convert this into a *Graphic* symbol, and name it *rectbutton*. Delete the instance from the stage.

Add a new blank symbol (**Insert** → **New Symbol**), set the behavior as *Movie Clip*, and call the symbol *fade-in*. You are now in symbol-edit mode. Drag an instance of the *rectbutton* graphic symbol you created before onto the stage. Align the symbol to the center of the stage. Insert a keyframe at frame 18, and create a *Motion Tween* between the first and eighteenth frames.

Click inside frame 1. Open the *Effect* panel (**Window** → **Panels** → **Effect**). In the first drop down list, select *Alpha*. Set the percentage to 30%. Move the play head back and forth to test the movie clip.

This movie clip will be used for a mouse over effect. Since we don't want it to play continuously, we would have to add a *Stop* action to the last frame. Add another layer to the *fade-in* movie, call it actions and insert a keyframe at frame 18. Right click on this frame, and select **Actions** from the context menu. Under *Basic Actions*, double click on the word *Stop*. The *Stop* action tells the movie to stop playing at a specified frame.

Add another blank symbol (**Insert** → **New Symbol**), set the behavior as *Movie Clip*, and call the symbol *fade-out*. Drag an instance of the *rectbutton* graphic symbol onto the stage. Align the symbol to the center of the stage. Insert a keyframe at frame 18, and create a *Motion Tween* between the first and eighteenth frames. Click inside frame 18, open the effect panel, select *Alpha* from the drop down list, and set the percentage to 30%. Move the play head back and forth the test the movie clip. Insert a layer, and call it actions. Insert a keyframe into frame 18, and add a *Stop* action to the frame. This movie clip will be used for a mouse out effect.

You now have three symbols in the *Library* – one graphic and two movie clips. Let's create the animated button. Select **Insert** → **New Symbol**. Name the symbol *mybutton*, and set its behavior to *Button*. Place an instance of the *fade-out* movie clip symbol in the *Up* frame, and align it to the center of the stage. Insert a blank keyframe in the *Over* frame. Drag an instance of the *fade-in* movie clip symbol onto the stage, and align it to the center of the stage. Insert a blank keyframe in the *Hit* frame. Drag an instance of *rectbutton* onto the stage, and align it to the center of the stage.

Enter the movie-editing mode by clicking on *Scene 1* in the upper left of the screen. Drag the *mybutton* symbol onto the stage, and align it to the center of the stage.

To view the movie, select **Control → Test Movie**.