

FLASH

CHAPTER 1

Introduction to Flash 4

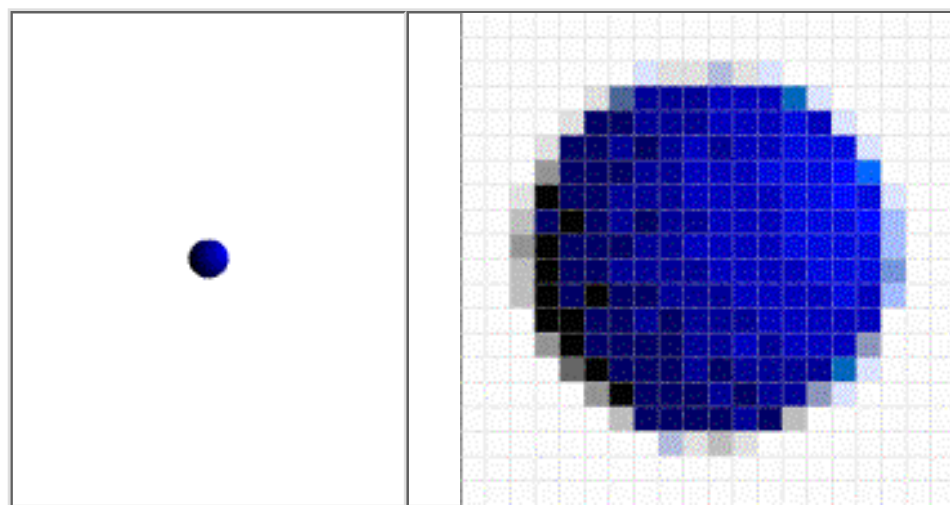


What is Flash?

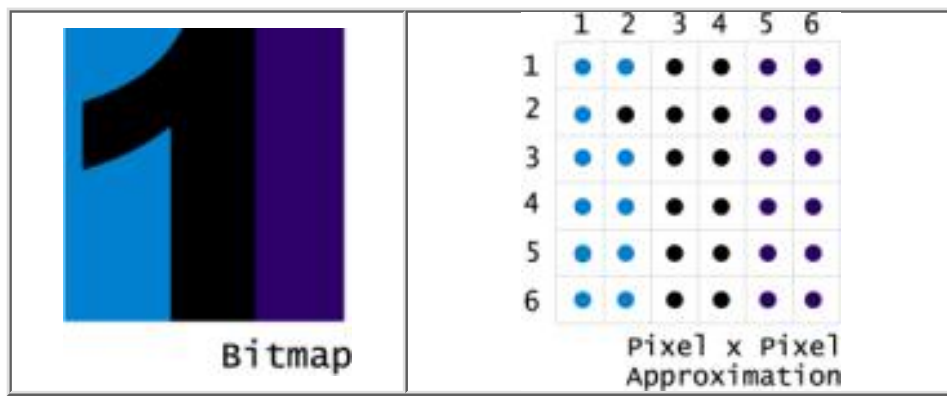
Flash is an authoring environment that gives web developers the power to create high-impact, fully interactive, animated web sites. Previous to Flash, creating animated web sites (complete with sound) involved large files, and often multiple files, which required equally robust bandwidth that the general public does not have. Flash files, in contrast, can be very small, making bandwidth less of a concern. Furthermore, Flash brings a level of advanced animation and interactivity previously unavailable using traditional Internet technology.

Vector Graphics

The key to Flash is its use of vector-based graphics. There are two primary digital image types: bitmaps and vectors. Bitmap images are the most common type of image file outside the professional print world. Bitmap files map out or plot the image pixel by pixel. Computer displays are made up of a collection of individual points or dots, called pixels, organized in columns and rows (similar to a table or a grid) where each square represents each pixel in the image. The image file basically plots each pixel, such as pixel 300 wide by 250 high is red. Pixel 301 wide by 250 high is also red. Pixel 302 wide by 250 high is yellow and so on. The larger the image, the larger the file size. Because bandwidth is a major concern when developing web sites, keeping the size of images to a minimum is very important. The main benefit of bitmap images is photographic color, tone, and texture is accurately produced.



This bitmap file is 20 pixels wide by 20 pixels high. On the left is the bitmap as it appears on the Internet; on the right is a magnified version. Notice the collection of pixels, from white surrounding the ball to various shades of gray.



If you were to plot a bitmap on a grid, each pixel is detailed. Essentially, this is what your computer does when it displays bitmaps.

Vector images do not plot images on a pixel-by-pixel basis. Instead, vector files contain a description of the image expressed mathematically. Essentially, the file tells the computer about the image and the computer draws it. For instance, the image of the circle in vector format, written in plain English, might look something like this:

- Draw a circle
- Radius is 2 inches
- Color is blue

A benefit of vector images is file size, particularly relative to the file size of bitmap images. The same circle as a bitmap file, in English might look something like this:

Row 1

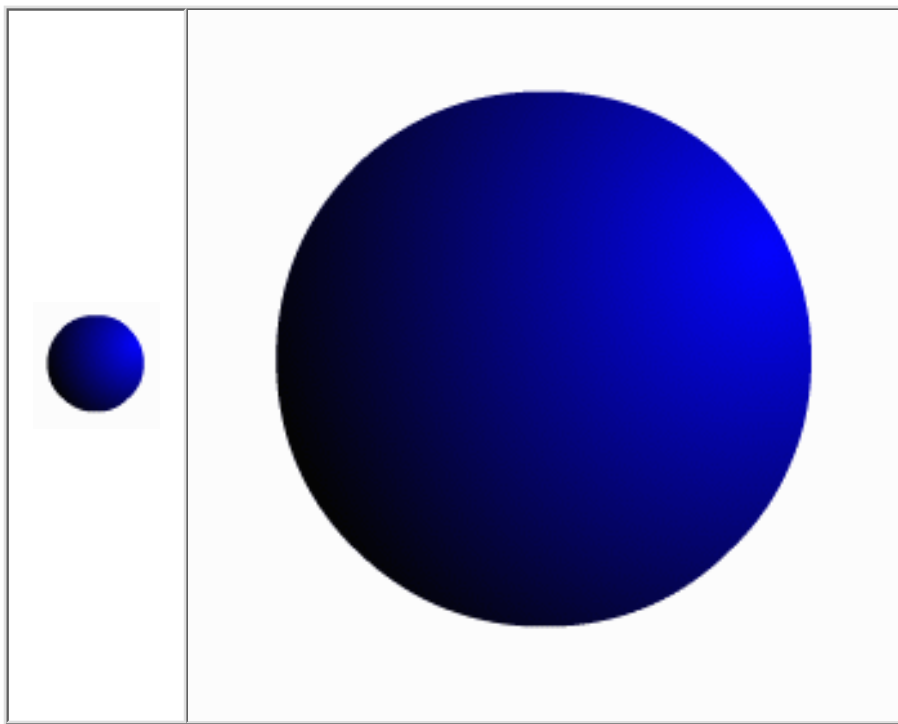
- Pixel 1x1 is white
- Pixel 2x1 is white
- Pixel 3x1 is white
- Pixel 4x1 is white, etc. until we reach Pixel 20x1

Row 2

- Pixel 1x2 is white
- Pixel 2x2 is white, etc.

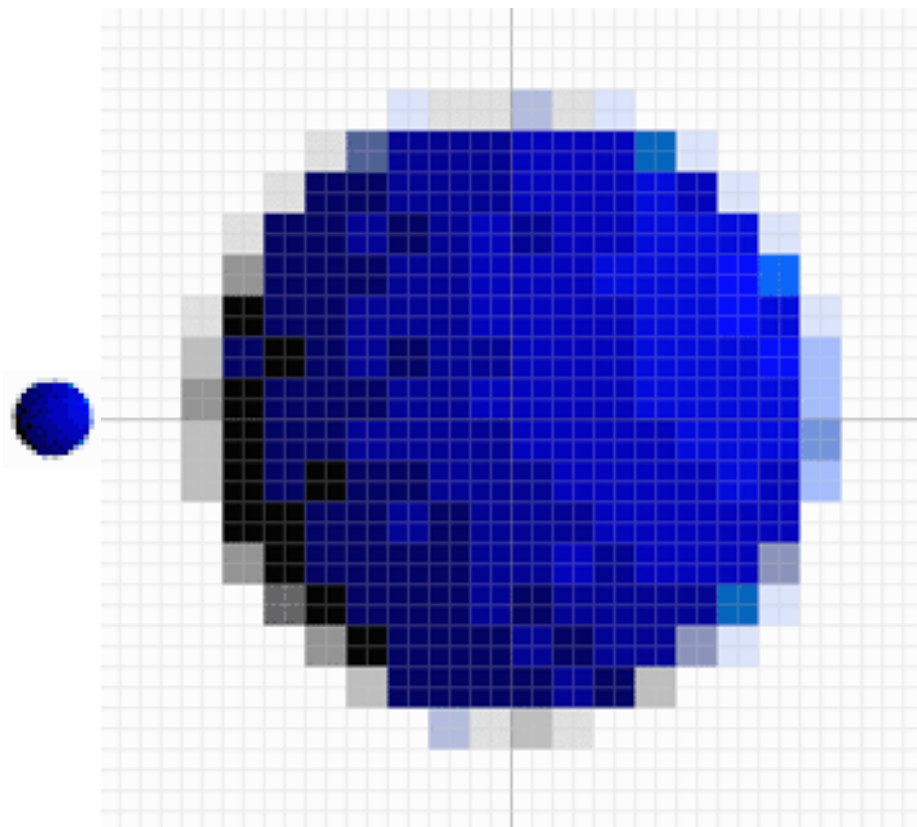
Describing the circle (and the color surrounding it) on a pixel-by-pixel basis is a much more lengthy process than telling the computer to draw the circle itself.

Because the computer effectively draws the image for us, vector graphics are also scalable, meaning the size of the image can be increased or decreased without degradation. Because the file is an equation, placing different numbers into the equation results in the computer rendering the image correctly.



Scaling vectors results in perfect images. The image on the left is the ball in its original size. The image on the right is the same ball with an increase in scale. The file size is the same.

Bitmaps are not scalable. If you increase the scale of a bitmap, the image degrades because the computer does not have enough information to create the new image. It merely takes the existing pixel information and translates one pixel into multiple pixels when you increase the scale of a bitmap engine. For instance, if we scale our 20x20 image to 40x40, each individual pixel in the image in its original state becomes four (4) pixels in its scaled state. This results in jagged edges.



Doubling the size of the bitmap causes the computer to replace each original pixel with four. As a result, the image becomes fuzzy or jagged. The file size also increases.

Unfortunately, vector images, because they describe images in terms of shapes, lines, curves, points, colors, length, etc., do not accurately reproduce photographic images. Vectors are limited to images with continuous tones, definite outlines and shapes

Internet Image Standards

In addition to two types of digital images (bitmaps and vectors), there are also a variety of bitmap and vector file types. Bitmap files, at the most basic level, are just as described, a map or plot of each individual pixel. However, image files can and often do contain other information depending on their use. For instance, the author or creator of the image might be included, as well as the date and copyright information. In addition, because bitmaps tend to be large, various methods of compression or methods to reduce file size have been developed.

To designate extra information and compression methods, different file types have been developed. File types designate a particular variety of bitmap or vector. Windows and Unix operating systems identify the file type by adding a period followed by three letters to the end of the filename, as in “circle.gif” or “circle.jpg.” This is called a file extension. On the Macintosh, file extensions are not used to designate file types. A file designating the file type and the application that created it is attached to the image file instead. However, Web servers and browsers also use file type extensions, so it’s advisable that Macintosh users do the same for image and Flash files generated for the web.

Web browsers currently support three image file types. They are .gif, or Graphics Interchange Format (GIF), .jpg or Joint Photographic Experts Group (JPEG), and .png or Portable Network Graphics (PNG). Each of these formats is a bitmap file. Currently, browsers do not support the display of vector-based images.

You can use each type of bitmap in Flash. However, for the most part, your use of bitmaps will be limited because bitmaps significantly increase file size. Using bitmaps in Flash is covered later.

Internet Production Issues

There are sets of common Internet production parameters that developers have to keep in mind. Typically, web sites are developed to the lowest common denominator in terms of technical specifications. The assumed display resolution is 640x480 pixels with a maximum of 256 colors. In terms of HTML, the current default is version 3.0. Furthermore, images and content are kept small because its assumed most users have a connection speed not greater than 28.8kbs.

Flash manages to circumvent many of the typical Internet production parameters. Because it uses vectors, the graphics are scalable, and can adjust to whatever resolution users are in. Flash’s color palette matches the 216 web-safe color palette that the browsers use, ensuring that colors are displayed properly. If the user has the appropriate Plug-in, HTML 3.0 elements support placing Flash files in web pages, and the compact size of Flash movies reduces bandwidth concerns. However, because vector based graphics require the computer to draw or render images in real-time, and because these images are in motion in Flash, processor speed becomes an issue. Complex animations, with

numerous objects in motion, can become processor intensive and may require faster processors to work properly. It's important to test your Flash movies on slower processors to gauge performance.

Review Questions

1. How do vector images differ from bitmap images?
2. Why does Flash require a Plug-in to display in a web browser?
3. How is processor speed a factor in the playback of Flash movies?

Summary

As a result of this chapter, you should be able to

- Define Flash
- Understand Internet image standards
- Understand the nature of Plug-ins
- Understand common Internet limitations

CHAPTER 2

The Flash Environment



This chapter details the Flash workspace. Tools, menus, the Stage, and Preferences are summarized. Each aspect of the Flash interface is examined in depth in later chapters. At this point, it's essential that you familiarize yourself with the workspace so you can begin to learn Flash.

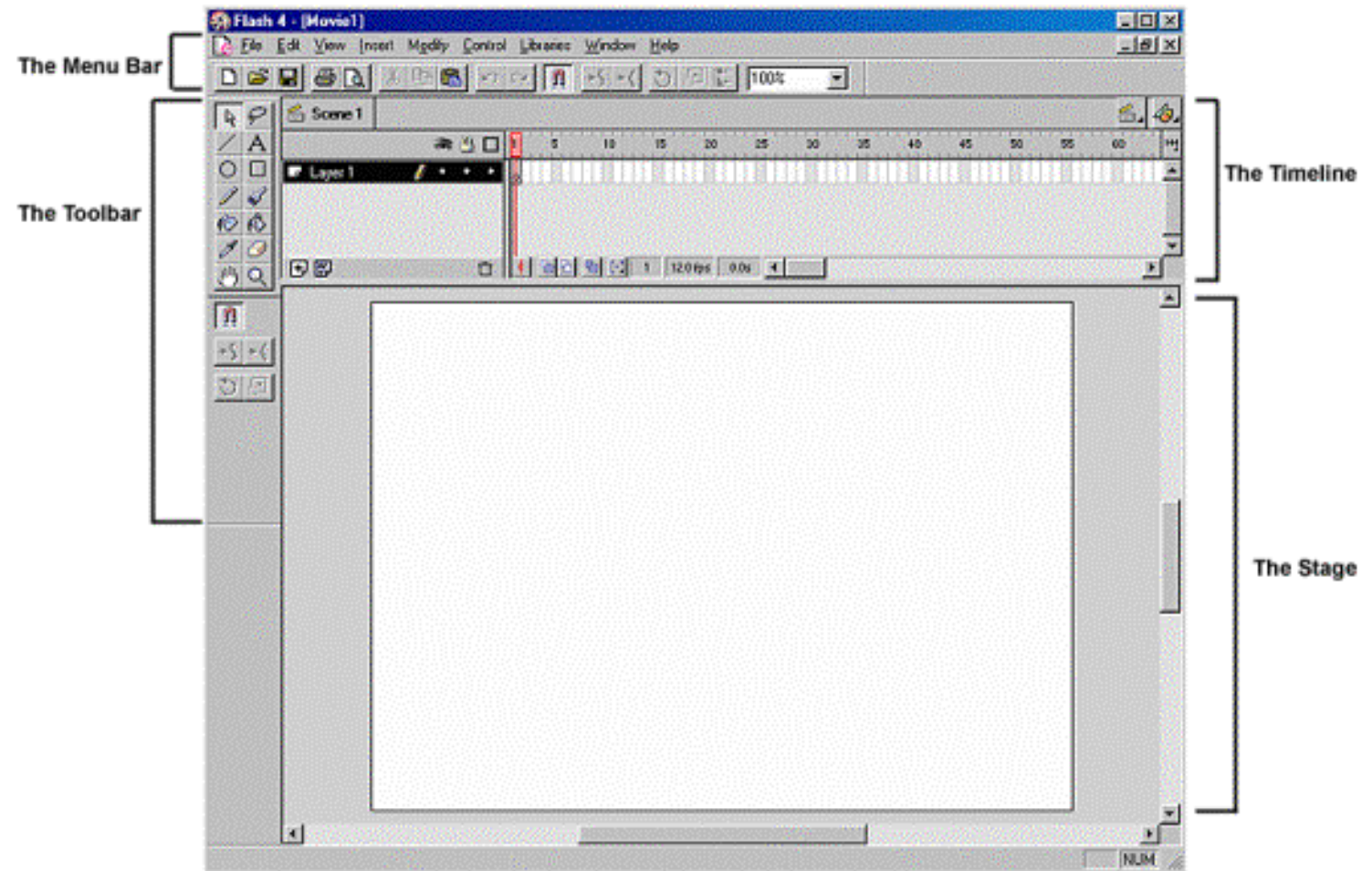
OBJECTIVES

Upon completing this section, you should be able to:

1. Understand the elements of the Stage
2. Understand the elements of the Timeline
3. Understand the Toolbar
4. Understand the Menu system
5. Understand Flash's Preferences

The Flash Workspace

The Flash workspace consists of four main areas: the Menu Bar, the Toolbar, the Timeline, and the Stage.



The Menu Bar

The Menu Bar contains controls for common functions such as opening and saving files, as well as specific functions, such as copying and pasting, calling up specific windows or palettes, and controlling the Flash environment. As you progress through each lesson, each of the Menus and Menu Items are detailed. The Menu Bar is divided into drop-down menus and a set of icons. The icons give you access to the most commonly accessed items.



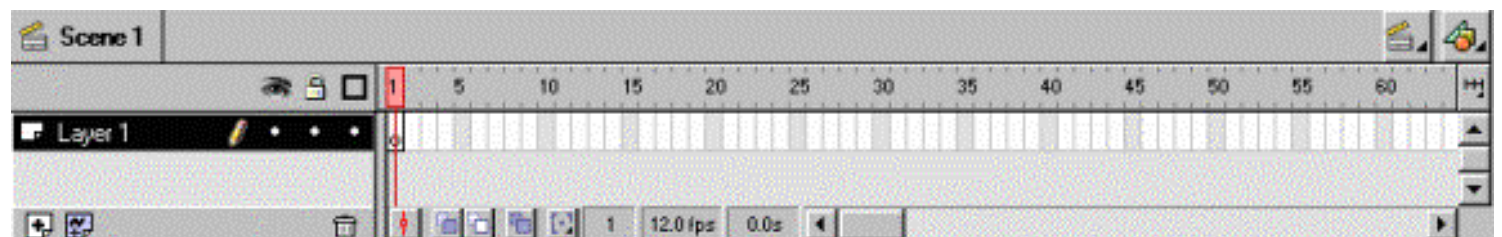
The Stage

The Stage is the area that you place content on and create interactive interfaces and animation. The gray area surrounding the Stage is called the “Work Area”, where you can also place content and animate it. However, users viewing your site only see the Stage. They cannot see content you place in the Work Area. In production, the Work Area is used when you need content to come from “off-screen” onto the Stage. Think of a stage production. Actors enter the stage from the left or right. In movies, elements can enter the frame from any one of four sides.



The Timeline

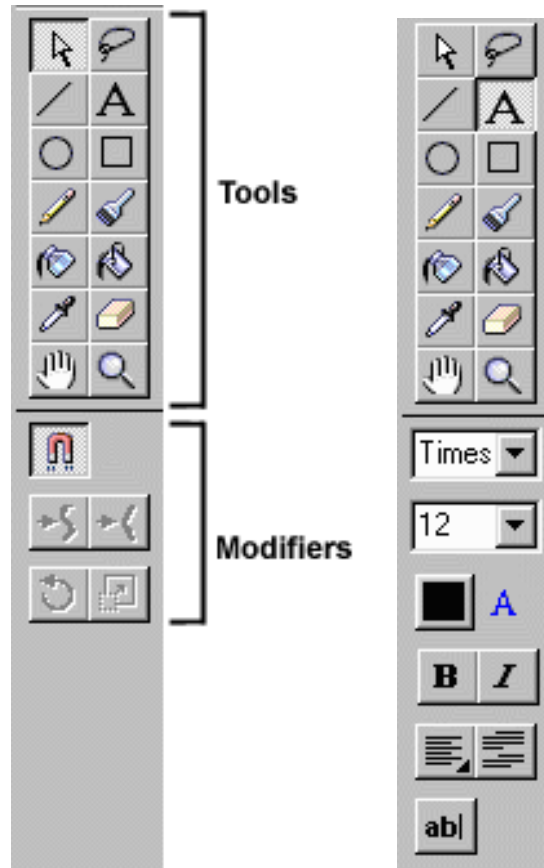
The Timeline manages the content of a Flash movie. You use the Timeline to build the layout of your Flash movie and animate the contents on the Stage. Think of the Timeline as a filmstrip and the Stage as a screen that displays each frame of the filmstrip in sequence, much like a movie theater. You will stage a sequence of events over time, just as a director stages a sequence of events over time in a movie. Furthermore, you use the Timeline to fine-tune or tweak your Flash movie, much like the editing process in traditional film where the editor assembles various pieces of the movie together, using time code to bring the pieces together at the right point in time. You'll find that timing in animation is very important.



The Toolbar

The Toolbar contains a collection of tools for creating and selecting content in the Timeline and on the Stage. The Toolbar is divided into two parts: Tools and Modifiers. Each Tool has a unique set of Modifiers that are displayed when you select the

Tool. For instance, when you select the Text Tool **A**, Modifiers such as Font, Font Size, Font Color, Font Style (Bold or Italic), Justification, and Text Field Values are made available.



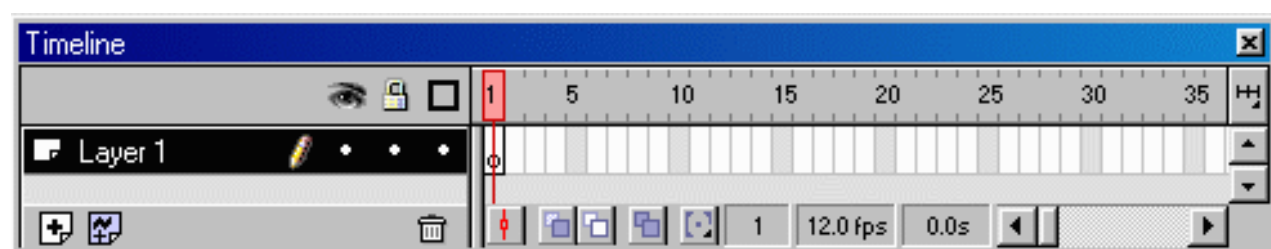
The Toolbar. In the first example, the Arrow Tool is selected. It has a single Modifier. In the second example, the Text Tool is selected. It has a number of Modifiers.

Customizing the Flash Workspace

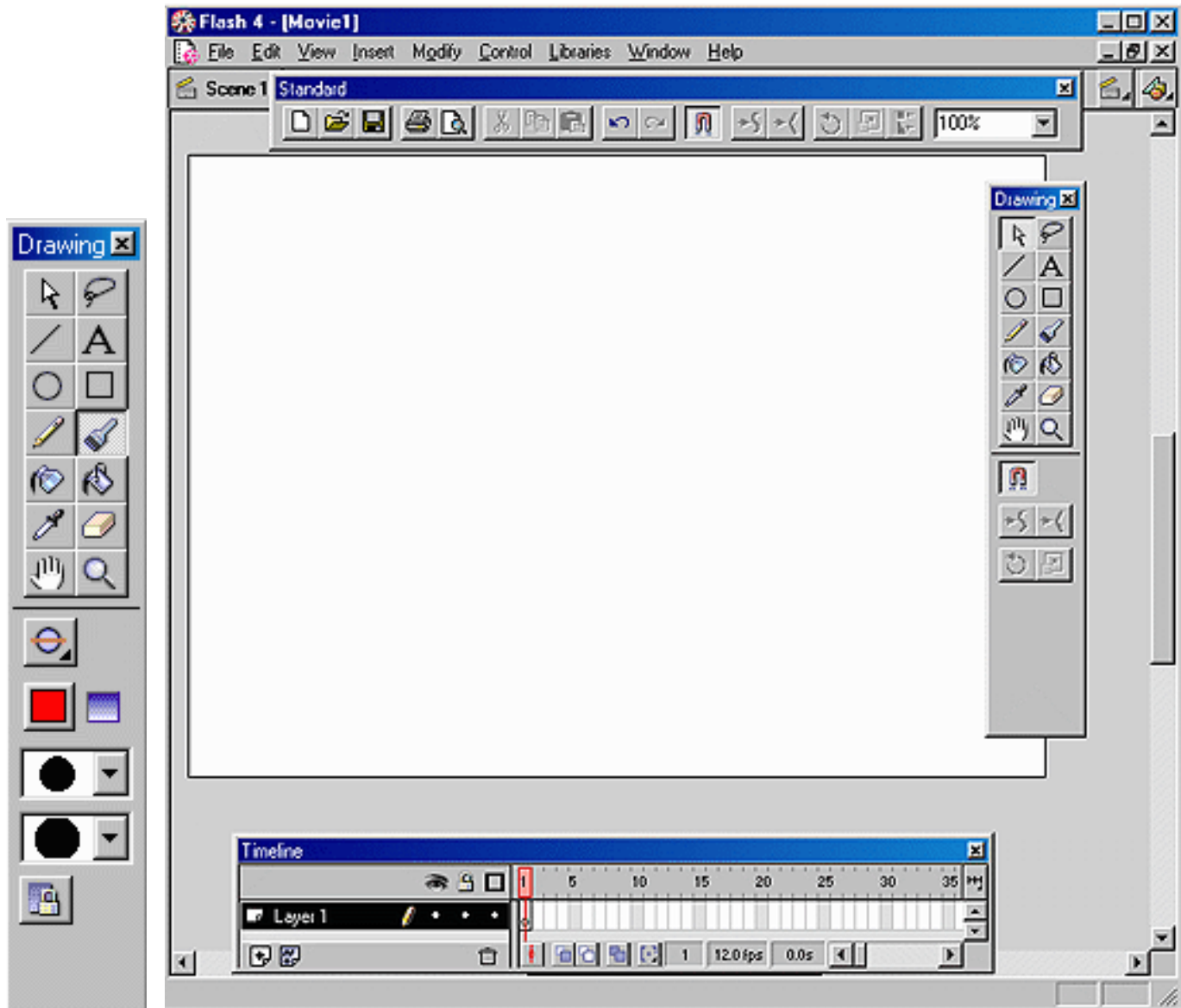
The Timeline, the Toolbar, and the icon set of the Menu Bar can be docked and undocked. Simply {Click}, hold and drag the item. Drop the item where you want to place it. The item is now undocked. To dock the item, simply {Click}, hold, and drag the item back to its original location.



The icon set of the Menu Bar undocked.



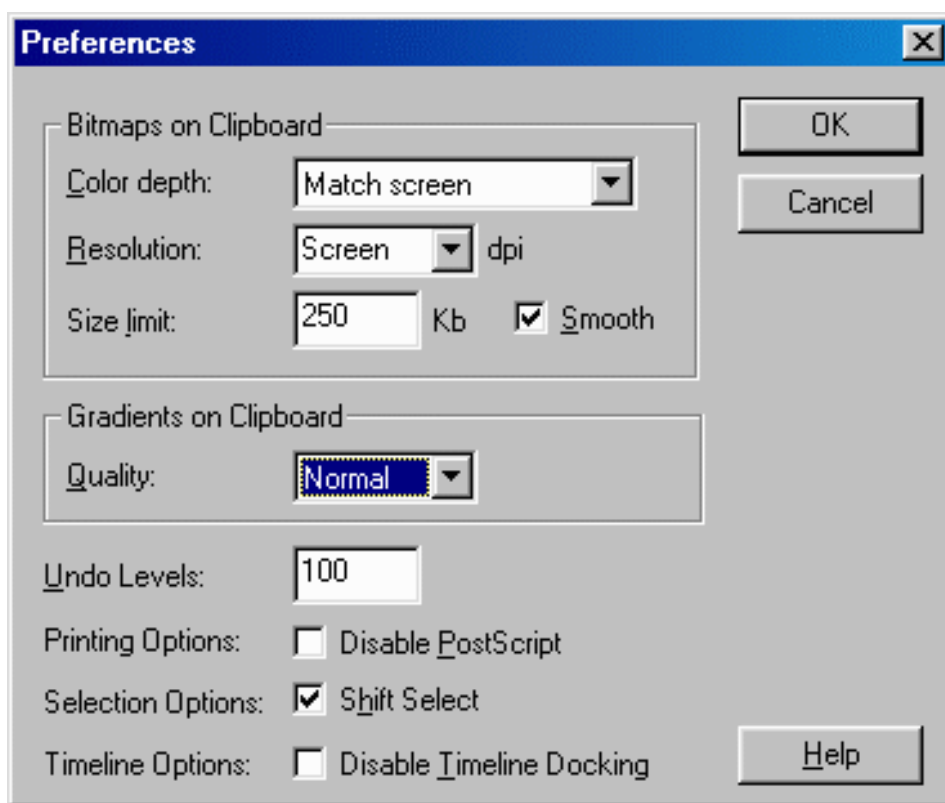
The Timeline undocked.



The Toolbar undocked. A customized Flash Workspace.

Preferences

Flash's Preferences include customizing the behavior of the Clipboard (the area where content is placed when you cut and paste), specifying the level of Undos, Printing Options, Selection Options, and Timeline Options. To access Flash's Preferences, select File/Preferences.



Bitmap and Gradient settings for the Clipboard are covered later. By default, Undo Levels is set to one hundred (100). Depending on the amount of memory you have, and how often you Undo operations during production, you may want to decrease or increase this amount.

In Printing Options, you can check Disable PostScript if you do not have a Postscript capable printer. Printing vector images with Flash is not the best method for print production. Dedicated print production applications, such as Macromedia Freehand, Adobe Illustrator, or Corel Draw are recommended. Flash is optimized for onscreen presentation.

In Selection Options, Shift Select is checked by default. This forces you to hold the [SHIFT] key and {Click} on different objects to make multiple selections. This is the standard way of making multiple selections, so we recommend keeping Shift Select checked.

In Timeline Options, Disable Timeline Docking is unchecked by default. This permits you to place the Timeline wherever you like on screen. However, many Flash developers like to keep the Stage and the Timeline together, in which case Disable Timeline Docking would be checked.

Review Questions

1. What are the main areas of the Flash workspace?
2. How are the Stage and the Timeline related?
3. What does dockable and undockable mean?

Summary

As a result of this chapter, you should be able to

- Understand the elements of the Toolbar
- Understand the elements of the Timeline
- Understand the Menu system
- Understand the Stage

CHAPTER 3

Flash Fundamentals: The Drawing Tools



You use Flash's Drawing tools to create image content for your movies. In this chapter, we detail each of the Drawing Tools and how you create image content or "objects" on the Stage. We recommend you follow along and try each of the tools to understand how they operate.

OBJECTIVES

Upon completing this section, you should be able to:

1. Understand the concept of the Shape Level in Flash
2. Use each of Flash's Drawing Tools
3. Understand the difference between Outlines and Fills

The Stage Level

In Flash, images you create or import can exist on various levels. Each level affects the behavior of the image and how you are able to modify it. The lowest, or first level is officially called the Stage level. Do not confuse the Stage level with the Stage; they are different things. To avoid potential confusion, we prefer to call the Stage level the Shape level, the level at which we draw, paint, and modify shapes (images can be referred to as shapes in Flash). The Drawing tools are only active when you are at the Shape level. Any shapes we create with the Drawing tools are created on the Shape level.

Shape Fundamentals

Before we begin to explore the Drawing tools, it's important to understand the fundamental logic behind shapes in Flash. Flash approaches shapes from two primary elements: outlines and fills. Each shape can consist of an outline and/or a fill. Flash's Drawing tools are divided into tools that create or modify outlines, and tools that create or modify fills.



The Line Tool

Use the Line Tool to create straight lines. *{Click}* the Line tool in the Toolbar to activate it. To draw lines, *{Click}*, hold, and drag the mouse pointer on the Stage. As you drag the mouse, a line extends from the point you *{Clicked}*. The line isn't actually drawn on the Stage until you *{Release}* the mouse button. To constrain the angle of the lines you draw to increments of 45 degrees, press and hold the [SHIFT] key as you draw the line.

Line Tool modifiers include line color, line thickness, and line style.



Line Color

To change the color of the line, *{Click}* the Line Color swatch. The Color Palette pops-up.

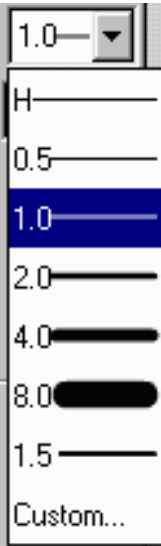


The color palette contains 228 color swatches. All the colors in the palette are web-safe, meaning they will display properly in a browser at an 8-bit color depth (limited to 256 colors). The web-safe color palette consists of 216 colors that both Internet Explorer and Netscape Navigator display. Flash's color palette has an additional 12 color swatches at the top row of the palette. These swatches are duplicates of commonly used web-safe colors. If you create custom colors, you should modify the colors

on the top row rather than modifying the web-safe colors. However, you may find the collection of colors on the top row useful, in which case you would add custom colors to the bottom of the palette. Creating custom colors is covered later in the chapter.

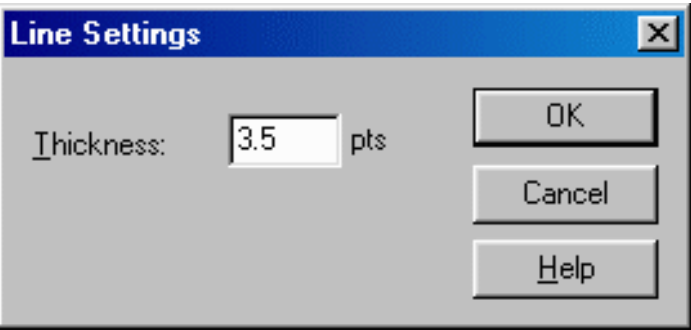
To select a color, {Click} on the color swatch of your choice. The color palette disappears and the newly selected color appears in the Line Color swatch.

Line Thickness

A vertical drop-down menu for line thickness. The top item is '1.0' with a small downward arrow to its right. Below it are several line segments of increasing thickness, each labeled with a value: 'H', '0.5', '1.0' (highlighted in blue), '2.0', '4.0', '8.0', '1.5', and 'Custom...'. The 'Custom...' option is at the bottom.

To change the Line thickness, {Click} the Line Thickness drop-down arrow and select the thickness you desire.

To create a custom line thickness, select “Custom...” from the drop-down menu. The Line Settings dialog-box pops up. Input the desired thickness. The custom line is added to the Line Thickness drop-down menu.

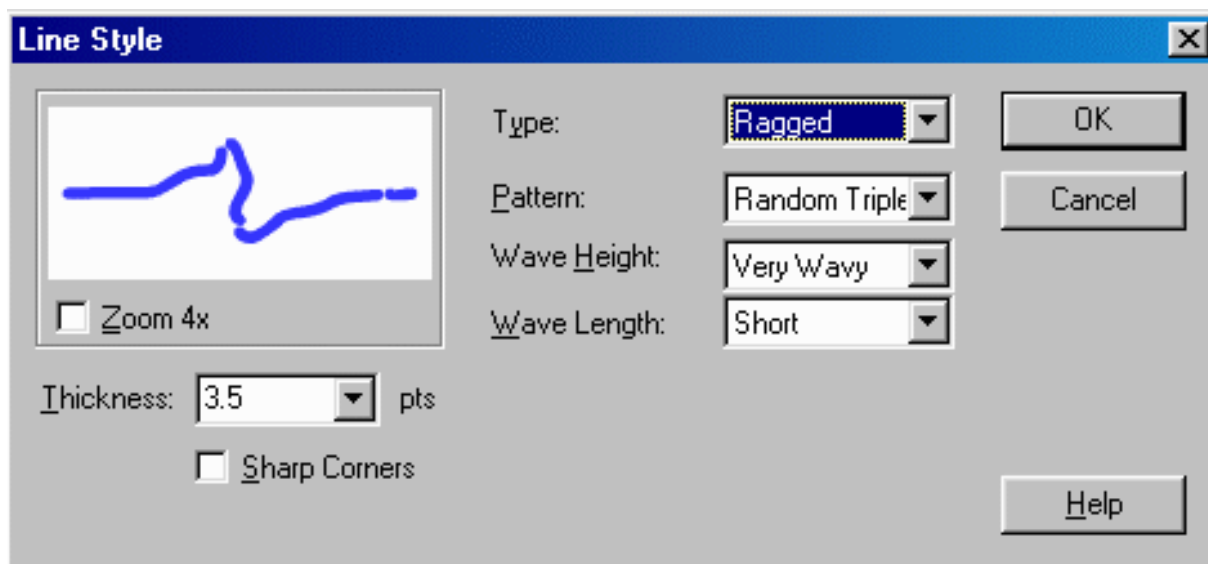


Line Style

A vertical drop-down menu for line style. The top item is 'Solid' with a small downward arrow to its right. Below it are several line segments of different styles: 'Solid' (highlighted in blue), a dashed line, a dotted line, a thick solid line, a thick dotted line, and 'Custom...'. The 'Custom...' option is at the bottom.

To change the Line style, {Click} the Line Style drop-down arrow and select the style you desire.

To create a custom line style, select “Custom...” from the drop-down menu. The Line Style dialog-box pops-up. Select from a range of options in “Type:”. Depending on what Type of line you select, a number of options are available to customize the line.



Experiment with the Line Tool. Try out different color, style, and thickness settings. When you're finished experimenting, select Edit/Select All from the Menu and press the [DELETE] key to clear the Stage.

The Oval Tool 

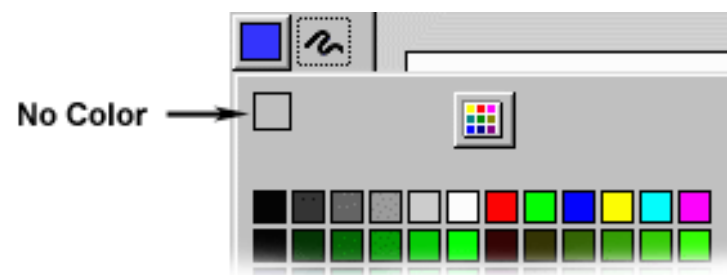
Use the Oval Tool to create circular shapes. {Click} the Oval Tool in the Toolbar to activate it. To create an oval (or circular shape), {Click}, hold, and drag the mouse pointer on the Stage. As you drag the mouse pointer, the circular shape is created. The circular shape isn't actually drawn on the Stage until you {Release} the mouse button. To constrain the circular shape to a perfect circle, press and hold the [SHIFT] button as you draw.

Oval Tool modifiers include Line Color, Line Thickness, Line Style, and Fill Color.



Line Settings

The set of Line Settings (Color, Thickness, and Style) operate in the same manner as the Line Tool with one exception. For Line Color, you can select No Color to make a circular shape without an outline around it. A colorless swatch is available in the color palette.



Fill Color

To change the Fill Color of the circular shape you create, *{Click}* the Fill Color swatch. The Fill Color Palette pops up. The Fill Color Palette resembles the standard Color Palette except there is a set of gradient fill colors available. Gradients are gradual shifts from at least one color to another color, or one color to transparency. The effect is similar to the way the sky shifts from dark blue to a lighter blue as you look toward the horizon.



The Fill Color Palette. In addition to the standard set of solid colors, there is a standard set of Gradient fill colors at the bottom of the palette.

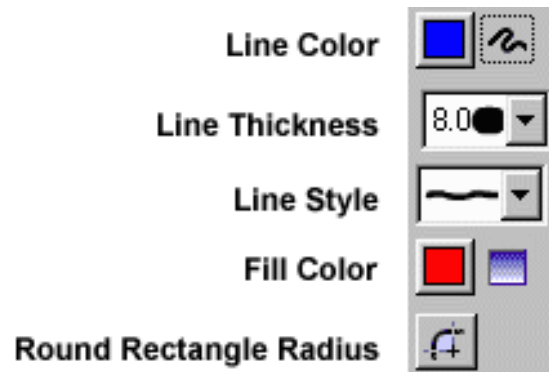
Select the fill color you want by *{Clicking}* on the color swatch.

Experiment with the Oval Tool. Try out different line color, style, and thickness settings as well as fill colors, both solid and gradient. When you're finished experimenting, select Edit/Select All from the Menu and press the [DELETE] key to clear the Stage.

The Rectangle Tool

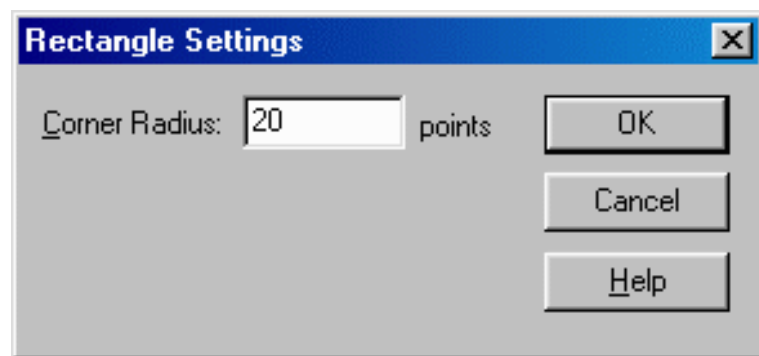
Use the Rectangle Tool to create rectangular shapes. {Click} the Rectangle Tool in the Toolbar to activate it. To create a rectangle, {Click}, hold, and drag the mouse pointer on the Stage. As you drag the mouse pointer, the rectangular shape is created. The rectangular shape isn't actually drawn on the Stage until you {Release} the mouse button. To constrain the rectangular shape to create a square, press and hold the [SHIFT] button as you draw the rectangular shape.

Rectangle Tool modifiers include Line Color, Line Thickness, Line Style, Fill Color, and Round Rectangle Radius.



Round Rectangle Radius

To round the corners of the rectangular shape, {Click} the Round Rectangle Radius modifier. The Rectangle Settings dialog-box pops up.



Input a value into the “Corner Radius:” field. Higher values increase the curvature applied to the corners of the rectangle.



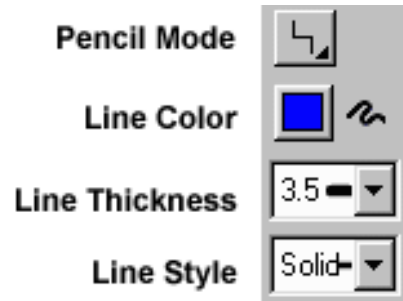
A rectangular shape with a corner radius of 20 points.

Experiment with the Rectangular Tool. Try out different line color, style, thickness settings, fill colors (both solid and gradient) and rounded corners. When you're finished experimenting, select Edit/Select All from the Menu and press the [DELETE] key to clear the Stage.

The Pencil Tool

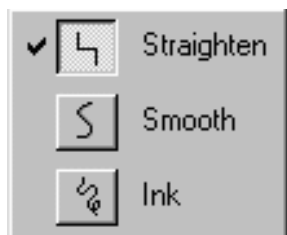
Use the Pencil Tool to draw freehand outlines (lines that aren't necessarily straight). {Click} the Pencil Tool in the Toolbar to activate it. To draw a line, {Click}, hold, and drag the mouse pointer on the Stage. As you drag the mouse pointer, the line is created. {Release} the mouse button to finish drawing the line.

Pencil Tool modifiers include Pencil Mode, Line Color, Line Thickness, and Line Style.



Pencil Mode

The Pencil Mode assists you as you draw your line, straightening or smoothing it out automatically. There are three modes: Straighten, Smooth, and Ink.



Straighten

Select “Straighten” to have Flash make the lines you draw straighten. Depending on how you draw the line, Flash will do its best to straighten each part. However, it does not create straight lines like the Line Tool.



The line at the left was drawn in Ink Mode (no assistance). The line at the right was drawn in Straighten Mode. Notice how Flash has automatically straightened each part of the line.

Smooth

Select “Smooth” to have Flash smooth out the lines you draw. Depending on how you draw the line, Flash will do its best to smooth each part.



The line at the left was draw in Ink Mode (no assistance). The line at the right was drawn in Smooth Mode.

Notice how Flash has automatically made the line very smooth.

Ink

Select “Ink” to turn off drawing assistance from Flash. The line you draw will be unaffected by Flash. Note however, that as you draw the line, the line you draw may look very rough until you {Release} the mouse button, at which point the true representation of the line occurs. It may appear that Flash is smoothing the line, but this is not the case.



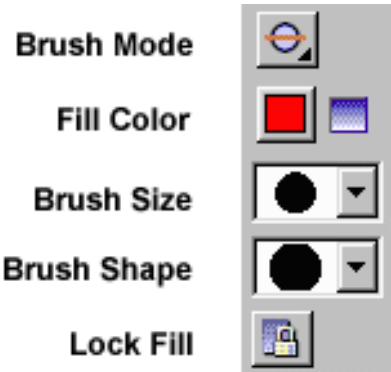
Drawing in Ink Mode before {Releasing} the mouse button.

The line in Ink Mode after {Releasing} the mouse button.

The Brush Tool 

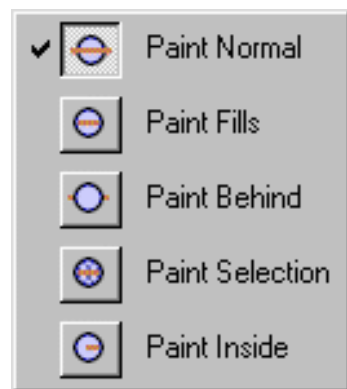
Use the Brush Tool to draw fills. {Click} the Brush Tool in the Toolbar to activate it. To draw a brush stroke, {Click}, hold, and drag the mouse pointer on the Stage. As you drag the mouse pointer, the stroke is created. {Release} the mouse button to finish drawing the stroke.

Brush Tool modifiers include Brush Mode, Fill Color, Brush Size, Brush Shape, and Lock Fill.



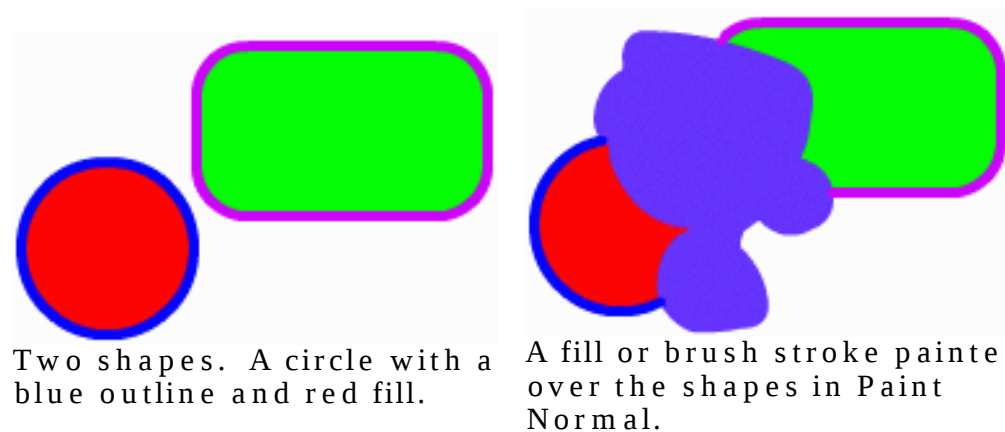
Brush Mode

Select from five Brush Modes to affect how your brush strokes behave in Flash. Modes include Paint Normal, Paint Fills, Paint Behind, Paint Selection, and Paint Inside.



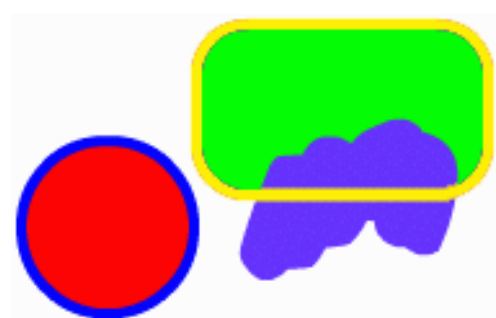
Paint Normal

Select Paint Normal mode to make your brush strokes paint over everything, without constraint. You can paint over any other shape on the Stage, whether it's an outline, fill, or both.



Paint Fills

Select Paint Fills to have the Brush paint only fills. It will not paint over outlines even though you can pass over the outlines while you make your paint strokes.

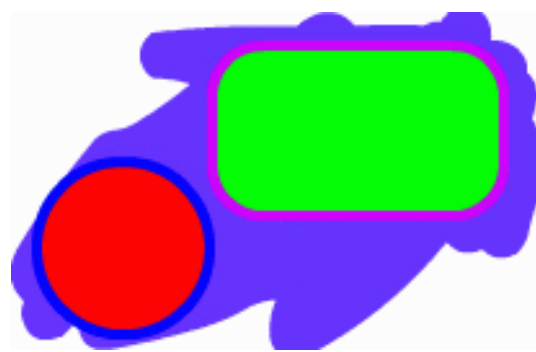


When using Paint Fills mode,

The Brush tool only paints fills, it ignores outlines.

Paint Behind

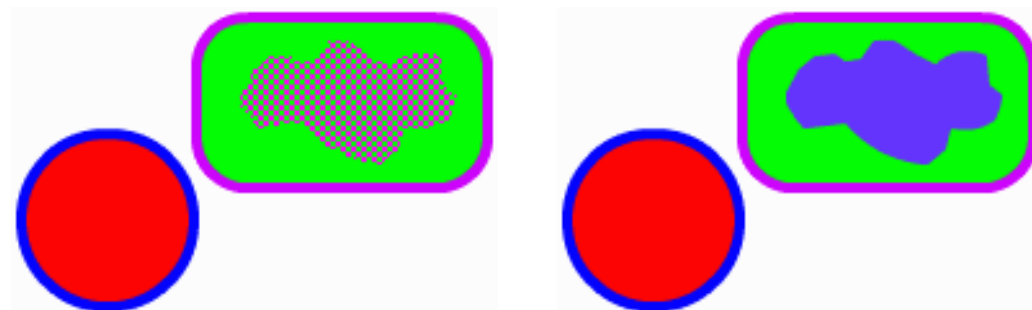
Select Paint Behind mode to have the Brush paint only behind existing shapes on the Stage. You are free to paint over the shapes, but when you *{Release}* the mouse button, only the area behind the shapes is painted.



With Paint Behind mode enabled, you can paint over shapes on the Stage. The fill color is automatically placed behind existing shapes when you *{Release}* the mouse button.

Paint Selection

Select Paint Selection mode to have the Brush fill only selected areas. Selections have not been covered yet. Basically, selections are areas of a shape you isolate from the rest of the shape by using one of Flash's selection tools. We cover selection tools later in the chapter. You make selections to affect changes to only the selected area. If you have an area of a shape selected, and you are using the Paint Selection mode of the Brush tool, only the selection is painted.

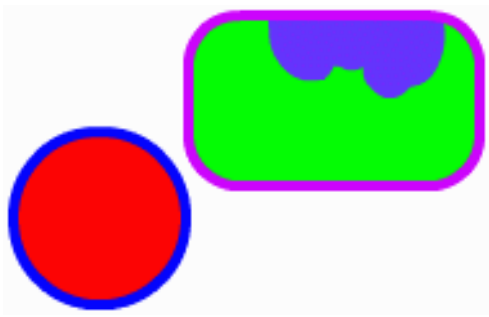


Part of the rectangular shape has been isolated by a selection.

With Paint Selection mode enabled, only the selected area is painted.

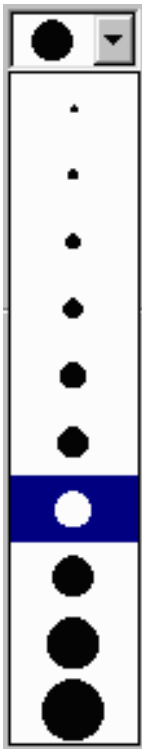
Paint Inside

Select Paint Inside mode to have the Brush paint only the areas inside of outlines. You must begin to paint within the outline. Even if you paint outside of the outline, having begun inside the shape, only the inside part of the shape is painted. The outline and the area outside of the outline are not affected.



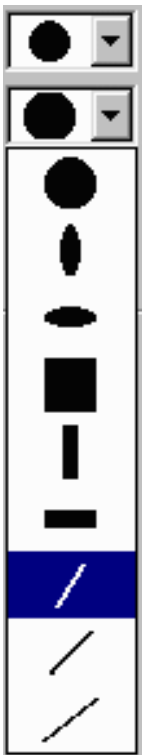
With Paint Inside mode enabled,
only the inside of a shape is painted.

Brush Size



To change the size of the Brush, select
a different brush size from the Brush
Size drop-down menu.

Brush Shape



To change the shape of the Brush,
select a different brush shape from
the Brush Shape drop-down menu.

Lock Fill

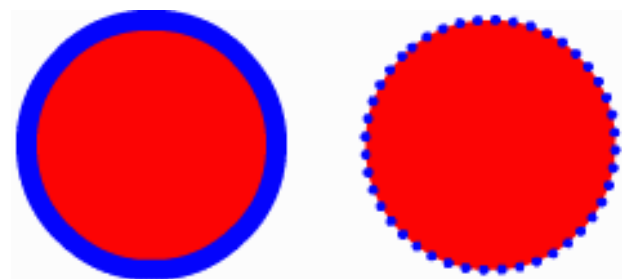
Lock Fill has to do with locking gradient fills and does not apply to the Brush Tool. We'll examine the function of Lock Fill later in the chapter when we cover the Paint Bucket Tool.

Experiment with the Brush Tool. Try out different fill colors, brush sizes, brush shapes, and brush modes. When you're finished experimenting, select Edit/Select All from the Menu and press the [DELETE] key to clear the Stage.

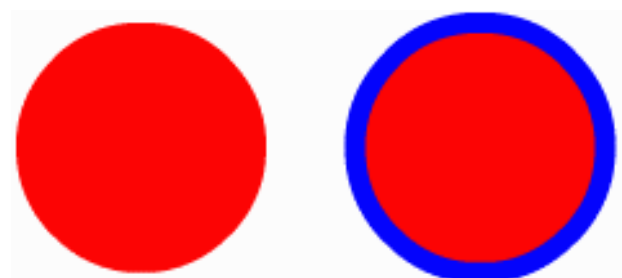
The Ink Bottle Tool

The Ink Bottle Tool modifies existing lines or adds lines to shapes without lines. To modify an existing line, select a line color, thickness, and style from the Ink Bottle modifiers and *{Click}* on an existing line. The line changes to the current settings you have selected in the modifiers. To add a line to an existing shape that does not have a line, select a line color, thickness, and style from the Ink Bottle modifiers and *{Click}* on the Shape.

Ink Bottle Tool modifiers include Line Color, Line Thickness, and Line Style.



The original shape (left), a circle with a red fill and a blue outline. Using the Ink Bottle, the outline is changed to a dotted line with a lower thickness (right).



The original shape (left), a circle with a red fill and no outline. Using the Ink Bottle, an outline is added to the shape (right).

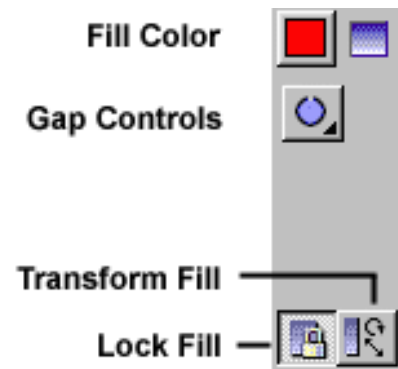
Experiment with the Ink Bottle Tool. Try out different line colors, line thickness, and line styles. Add lines to shapes without lines and modify existing lines. When you're finished experimenting, select Edit/Select All from the Menu and press the [DELETE] key

to clear the Stage.

The Paint Bucket Tool

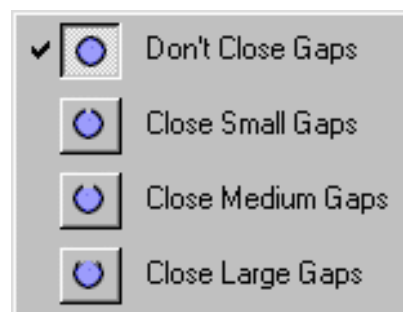
The Paint Bucket Tool modifies existing fills or adds fills to shapes without fills. To modify an existing fill, select a fill color (solid or gradient) and *{Click}* on the fill. To add a fill to an existing shape made only of an outline, select a fill color (solid or gradient) and *{Click}* inside the outline of the shape.

Paint Bucket Tool modifiers include Fill Color, Gap Controls, Lock Fill, and Transform Fill.

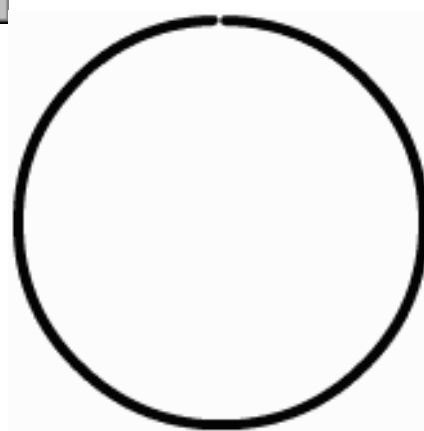


Gap Controls

When a shape without a fill has an outline that is not closed (such as a circular line with ends that do not meet) the Paint Bucket may have trouble creating a fill. Because the outline has a gap in it, it doesn't provide enough information for the Paint Bucket to create a fill within the outline's boundaries. To fill shapes with gaps in the outline, use the Gap Controls modifier. Controls include Don't Close Gaps, Close Small Gaps, Close Medium Gaps, and Close Large Gaps.



The range of Gap sizes, from Small to Large, are actually very small. For instance, the following outline is considered a Medium Gap by Flash.



With Close Small Gaps selected, the Paint Bucket is unable to fill this shape. You must select Close Medium Gaps. If you find that the Gap Controls

aren't functioning, keep in mind that Large Gaps are actually quite small in Flash.

Lock Fill

When you select a gradient as your fill color, the center point of the gradient is placed where you *{Click}* the Paint Bucket in the shape. Think of the center point of the gradient as an indicator of a virtual light source.



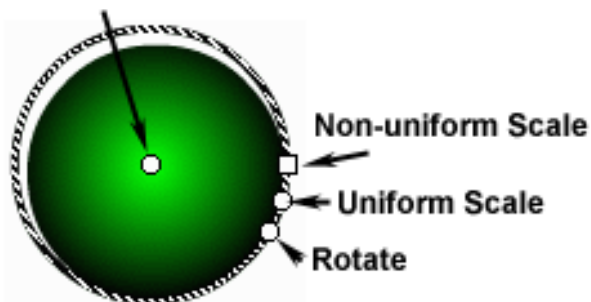
(From left to right) Gradients with a center point in the center of the shape, at the top right, and at the bottom left.

Once you have created a gradient fill, if you *{Click}* the Lock Fill icon, the center point (or if you like, light source) of the gradient is locked. You can still fill the shape with different gradient colors; however, you cannot change the center point when Lock Fill is selected.

Transform Fill

Transform Fill permits you to modify gradient fills. *{Click}* the Transform Fill icon and then *{Click}* the shape (which must have a gradient fill) that you wish to transform. The Transform Fill modifier surrounds the shape.

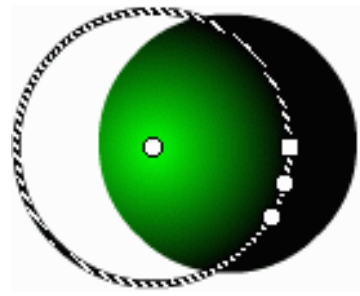
Center Point



The Transform Fill modifier consists of four controls; Center Point, Non-uniform Scale, Uniform Scale, and Rotate. *{Click}* the control of your choice, hold, and drag to transform the gradient fill. *{Release}* to make the change.

Center Point

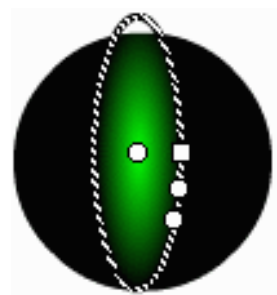
Use the Center Point control to move the center point of the gradient fill. Note that even if you have Lock Fill enabled, you can still move the center point with the Transform Fill modifier.



The Center Point moved to the left of the shape.

Non-uniform Scale

To compress or expand the gradient, use the Non-uniform Scale control.



The gradient fill is compressed using the Non-Uniform Scale control.

Uniform Scale

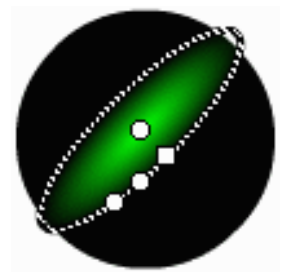
Use the Uniform Scale control to increase or decrease the scale of the gradient fill.



The gradient fill is reduced in scale using the Uniform Scale control.

Rotate

Use the Rotate control to rotate the gradient.



The gradient fill is compressed and then rotated using the Rotate control.

Experiment with the Paint Bucket Tool, creating shapes with outlines and without outlines and place fills in them. Try out the Lock Fill and transform Gradient fills with the Transform Fill modifier. When you're finished experimenting, select Edit/Select All from the Menu and press the [DELETE] key to clear the Stage.

The Dropper Tool 

Use the Dropper Tool to sample colors from existing shapes by *{Clicking}* on a fill or outline. When you sample an outline, the outline color swatches for all the outline tools change to the sampled color. Likewise, when you sample a fill, the fill color swatches for all the fill tools change to the sampled color. To make both the outline and fill color the same from a sampled color, hold the [SHIFT] key when sampling the color.

The Dropper Tool does not have any modifiers.

Experiment with the Dropper Tool. Create shapes with different colored outlines and fills. Use the Dropper Tool to sample colors. Cycle through each tool to see how the color swatches for outlines and fills are automatically changed to the color you've just sampled. When you're finished experimenting, select Edit/Select All from the Menu and press the [DELETE] key to clear the Stage.

The Eraser Tool 

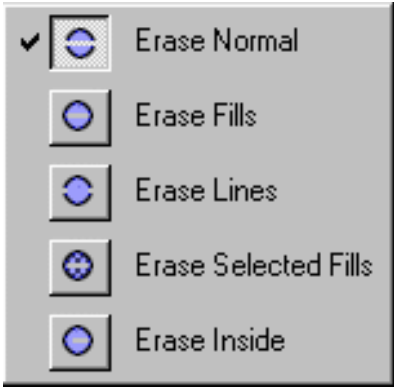
Use the Eraser Tool to erase shapes or parts of shapes on the Stage. To erase a shape (or part of a shape), *{Click}*, hold, and drag the mouse pointer over the shape.

Eraser Tool modifiers include Erase Mode, Faucet, and Erase Shape.



Erase Mode

Select from five Erase Modes to affect how the Eraser behaves in Flash. Modes include Erase Normal, Erase Fills, Erase Lines, Erase Selected Fills, and Erase Inside.



Erase Normal

Select Erase Normal mode to make the Eraser Tool erase everything, without constraint. You can erase any shape on the Stage, whether it's an outline, a fill, or both.

Erase Fills

Select Erase Fills to only erase fills. Outlines will not be erased, even if you pass over outlines with the Eraser Tool.

Erase Lines

Select Erase Lines to erase outlines only. Fills will not be erased, even if you pass over Fills with the Eraser Tool.

Erase Selected Fills

Select Erase Selected Fills to erase selections you have made. Fills and outlines will not be erased, even if you pass over them.

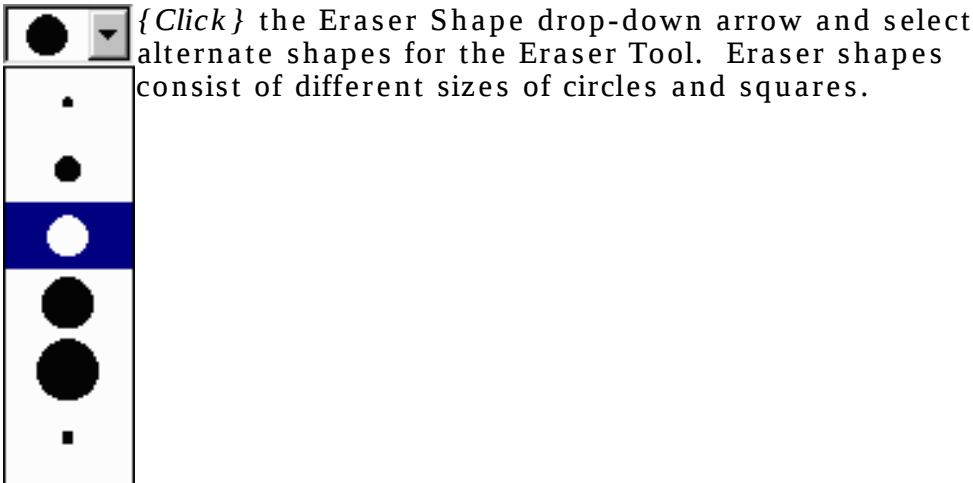
Erase Inside

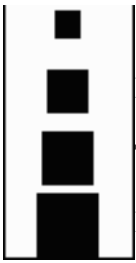
Select Erase Inside to only erase areas within an outline. You must begin to erase within the outline. Even if you erase outside of the outline, having begun inside the shape, only the inside part of the shape is erased. The outline and any shapes outside the area of the outline are not affected.

Faucet

The Faucet acts as quick eraser. With a single *{Click}*, an entire fill or outline is erased instantly.

Eraser Shape

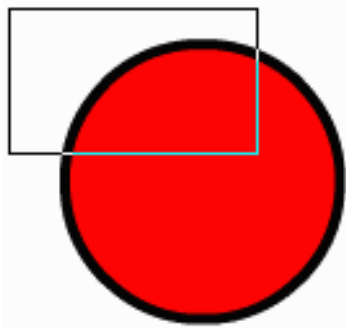




Experiment with the Eraser Tool. Create shapes with outlines and fills. Use the Eraser Tool to erase shapes. Try out the Erase Modes, the Faucet, and different Eraser Shapes. When you're finished experimenting, select Edit/Select All from the Menu and press the [DELETE] key to clear the Stage.

The Magnifier Tool

Use the Magnifier Tool to zoom in to and out of areas on the Stage. There are two methods you can use. By default, the Magnifier Tool will have a positive sign “+” in the lens. This indicates that zooming in (or increasing magnification) is enabled. Simply {Click} on the Stage and the Magnifier Tool will increase magnification in steps of 200%, 400%, and 800%. To decrease magnification, or to zoom out, simply press and hold the [ALT] key as you {Click} on the Stage. The Magnifier Tool will have a negative sign “-“ in the lens. Alternately, to increase magnification, you can {Click}, hold, and drag on the Stage with the Magnifier Tool. As you drag, a box is created indicating the area of the Stage that will be magnified when you {Release} the mouse button.



The area of the Stage magnified.

Drag to create the area of magnification and {Release} the mouse button.

Magnifier Tool modifiers include Zoom In and Zoom Out.



By default, the Magnifier Tool is set to increase magnification. {Click} the Zoom Out icon to change from the default.

Zoom Control

You can also use the Zoom Control drop-down menu in the Menu Bar. Simply {Click} the drop-down area and select a level of magnification.

Experiment with the Magnifier Tool and the Hand Tool. Create shapes with outlines and fills. Zoom into them using the Magnifier Tool and move the Stage around using the Hand Tool. When you're finished experimenting, select Edit/Select All from the Menu and press the [DELETE] key to clear the Stage.

The Hand Tool



Use the Hand Tool to move the Stage around within the Flash environment. At some point, during production, you may have zoomed in or magnified a particular area of the Stage to modify a shape. You can use the Hand Tool to move the Stage around in the magnified view instead of the scroll bars. Simply select the Hand Tool and *{Click}*, hold, and drag the Stage. *{Release}* the mouse button when you've moved the Stage area to the desired location.

The Hand Tool does not have any modifiers.

Review Questions

1. Every shape in Flash can be created with two primary elements. What are they?
2. List some of the properties of Flash's color palette.
3. How do you round the edges of a rectangular shape?
4. When using the Paint Brush, what are some of the modifiers available to aid in creating shapes?

Summary

As a result of this chapter, you should be able to

- Understand the concept of the Shape Level in Flash
- Use each of Flash's Drawing Tools
- Understand the difference between Outlines and Fills

CHAPTER 4

Flash Fundamentals: Selections, Text, and Gradients



Understanding and using Flash's Selection Tools is essential in creating shapes and editing animation. This chapter introduces you to Flash's various selection tools and methods, as well as detailing the use of text, and the creation of custom colors and gradients.

OBJECTIVES

Upon completing this section, you should be able to:

1. Use each of Flash's Selection Tools
2. Use Selection Tool Modifiers
3. Understand Text in Flash
4. Understand Gradient fills and how to create new Gradients

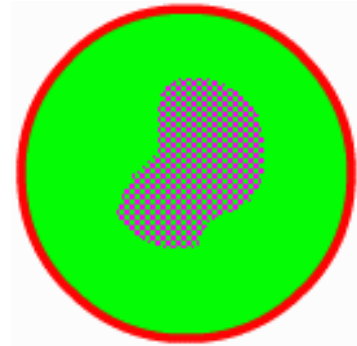
Selection Tools

An integral part of creating shapes in Flash is making use of the Selection Tools. Selections are areas of a shape you isolate from the rest of the shape to make changes to that particular area. Flash has two selection tools, the Lasso Tool and the Arrow Tool.

The Lasso Tool



Use the Lasso Tool to create selections. *{Click}*, hold, and drag the mouse to make your selection. Simply draw, in freehand, around the area you want to make into a selection, similar to drawing with the pencil tool in Ink mode. When you *{Release}* the mouse button the selection is made. A checkered selection pattern indicates the area that's selected.



The Selection pattern in the circle shape.

Lasso Tool modifiers include Magic Wand, Magic Wand Properties, and Polygonal Mode.

Magic Wand



Magic Wand Properties

Polygonal Mode

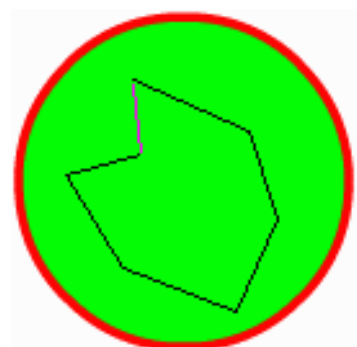


Magic Wand

The Magic Wand and Magic Wand Properties apply to selecting bitmaps within Flash, not standard vector shapes. We cover their use in another chapter.

Polygonal Mode

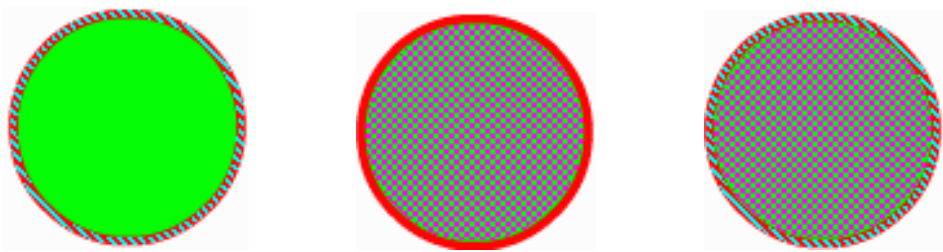
Enabling Polygonal Mode forces the Lasso tool draw in straight lines. You *{Click}* and *{Release}* the mouse button to create the start point of the line. As you drag the mouse the first line is drawn, *{Click}* the mouse to make the end point of the first line and the beginning point of the new line. Continue *{Clicking}* and dragging until you have made your selection, often in the shape of a polygon (hence the name of the tool). To complete the selection, you have to *{Click}* the starting point again. *{Double-Clicking}* at any time forces Flash to draw a straight line from whatever location you are in to the start point of the selection.



A Polygonal Mode selection.

The Arrow Tool

Use the Arrow Tool to make selections and modify shapes. When you make selections with the Arrow Tool you simply have to {Click} once to make the selection. If you {Click} an outline, the entire outline is selected. If you {Click} a fill, the entire fill is selected. {Double-Clicking} a shape made of both outlines and fills will select both the outline and fill.



Outline Selected Fill Selected Outline and Fill Selected

You can modify a selection with the Drawing Tools, or move the selection around the Stage.

If you {Click}, hold, and drag on an outline with the Arrow Tool, you can modify the shape of the outline. The outline will curve if a small arc appears below the Arrow Tool on the Stage.



Original Line. Line is curved using the Arrow Tool.

When you {Click} and hold the end of an outline, a small circle appears (indicating a point in the line.) You can drag the point as well to modify the outline. When you grab an end point, the mouse pointer displays a right-angled line.



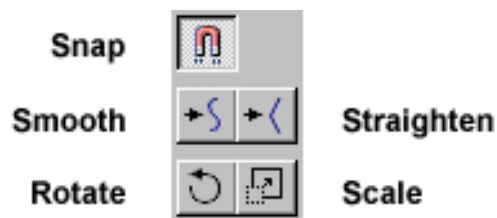
The endpoint is grabbed with the Arrow Tool. The Line is curved.

You can {Click}, hold, and drag the edges of fills to modify their shapes as well, just as you do with outlines.



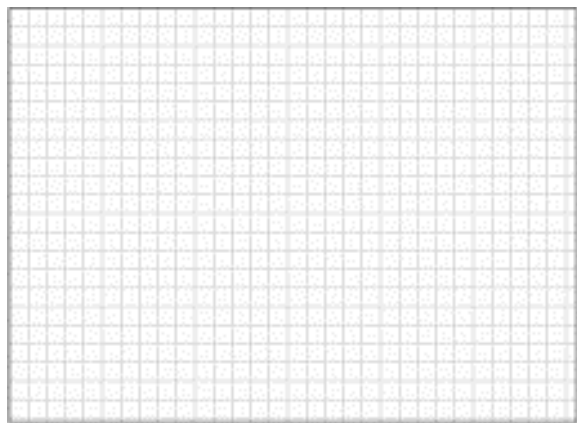
Original Fill. Modified by the Arrow Tool.

Arrow Tool modifiers include Snap, Smooth, Rotate, Straighten, and Scale.



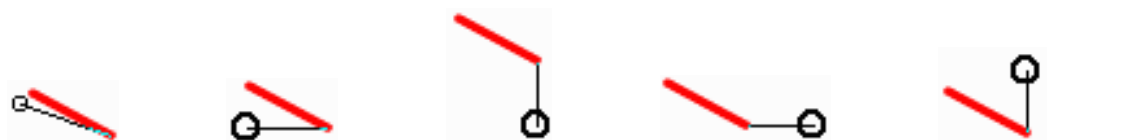
Snap

Snap is enabled by default. There is a grid on the Stage (invisible by default), which shapes snap too, meaning they are attracted to the nearest grid line like a magnet is attracted to metal. To view the grid, select View/Grid from the Main Menu.



The Stage with the Grid made visible.

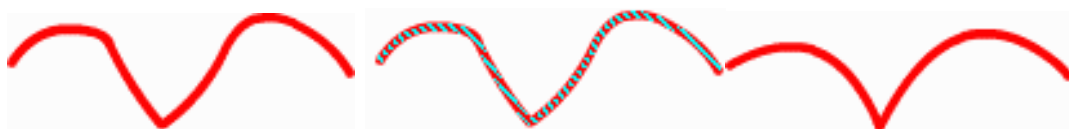
The effect is subtle in Flash and you may not have noticed it. We cover setting Snap and Grid settings later. The best way to see how Snap functions is to grab the endpoint of a line and move it around in a circle. You'll notice that the end point indicator (the hollow circle) increases in size and the line snaps or is pulled to 90-degree angles. When you move entire shapes, the center of shape snaps to the invisible grid as well.



As you move the endpoint around it snaps at 90-degree angles with Snap enabled.

Smooth

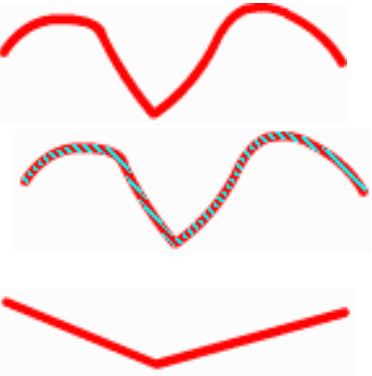
If you have drawn an outline and you are not satisfied with how smooth it is, you can select it with the Arrow Tool and *{Click}* on the Smooth modifier to smooth the line. Repeatedly *{Clicking}* the Smooth modifier makes the line smoother.



An outline is selected and smoothed out with the Smooth modifier.

Straighten

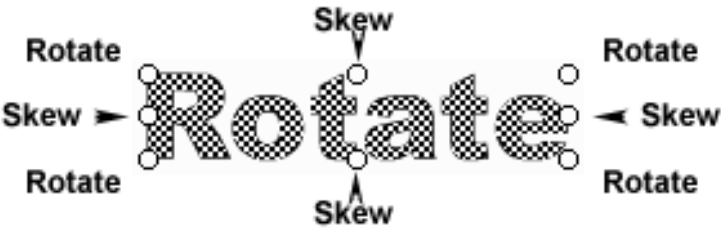
If you have drawn an outline with curves in it and you want to straighten it, select the outline with the Arrow Tool and {Click} the Straighten modifier. Repeatedly {Clicking} the Straighten modifier will eventually make the outline completely straight.



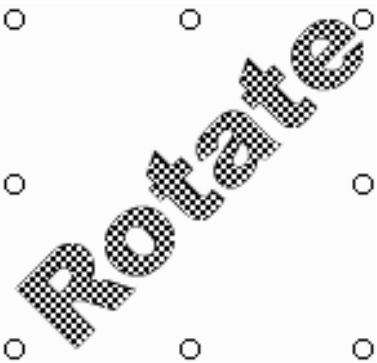
An outline is selected and straightened with the Straighten modifier.

Rotate

To rotate a shape, select the shape with the Arrow Tool, {Click} the Rotate modifier, and {Click} and drag a handle on the Rotation modifier box to the degree you wish to rotate the shape. Rotation handles are at each corner of the box. Skewing handles are between the Rotation handles.



A Rotation box around a shape.



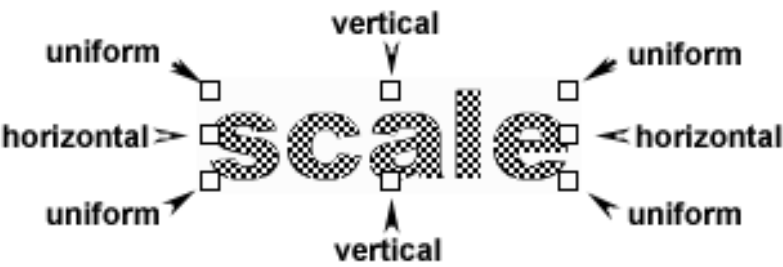
The shape rotated.



A shape skewed to the left.

Scale

To scale a shape, select the shape with the Arrow Tool, *{Click}* the Scale modifier, and *{Click}* and drag a handle on the Scale modifier box to increase or decrease the scale of the shape. To scale the object uniformly, use the handles at each corner of the box. To scale the object on the horizontal or vertical access, use the horizontal or vertical handles appropriately.



A Scale box around a shape.



A shape scaled along the horizontal access.



A shape scaled along the vertical axis.



A shape scaled uniformly (decreased).

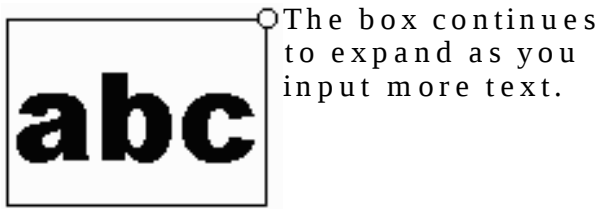
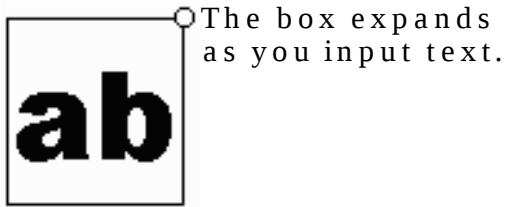
Experiment with the Lasso Tool and the Arrow Tool. Create shapes with outlines and fills. Use the Selection Tools to create selections and modify the shapes, either by changing their shape, or by modifying fill and outline properties (color, line styles, etc.). When you're finished experimenting, select Edit/Select All from the Menu and press the [DELETE] key to clear the Stage.

The Text Tool 

Use the Text Tool to create text in Flash. There are two methods of creating text on the Stage. You can select the Text Tool and *{Click}* on the Stage. A text box appears with a small hollow circle at the top right. As you input text, the box expands with each character (letter) you input.



Expanding Text Box.



Alternately, you can select the Text Tool and *{Click}*, hold, and drag the mouse cursor. As you drag the mouse, a text box is created. When you *{Release}* the mouse button, a text box with a hollow square at the top right is on the Stage. This box is a Paragraph Text box. As you input text, the text wraps to the next line when you reach the edge of the box.

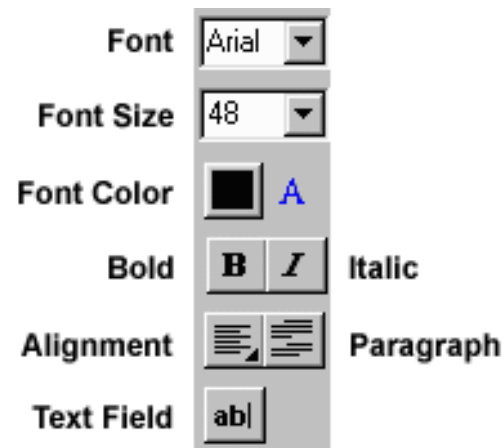


Paragraph Text Box.



As you input text, the text wraps to the next line when it hits the edge of the Paragraph box.

Text Tool modifiers include Font, Font Size, Font Color, Bold, Italic, Alignment, Paragraph, and Text Field.



Font

{Click} the Font drop-down arrow to select the font (also called typeface) you desire. Flash lists all the fonts currently installed on your computer. You might notice that the edges of text are not smooth. Don't worry. Smoothing the edges of text in production causes Flash to operate slowly (depending on the speed of your computer). When you actually export the Flash movie for the Internet, text edges will be smooth. However, in some cases, because not all fonts are created equal, some fonts cannot be smoothed. To test whether or not a font can be smoothed, select View/Antialias Text from the Menu Bar. If the text you input is smooth then the font is valid. If not, then you shouldn't use that particular font. Make sure you select View/Antialias after you have checked whether or not the font you've selected is valid. Why? Because it will decrease the overhead on your processor as you produce Flash movies.

Font Size

{Click} the Font drop-down arrow to select preset Font Sizes, or input a value manually in the Font Size drop-down box. Keep in mind that you can scale the text afterward if you need to.

Font Color

{Click} the Font Color swatch to select a color for the text you create.

Bold

{Click} the Bold modifier to bold your text.

Italic

{Click} the Italic modifier to italicize your text.

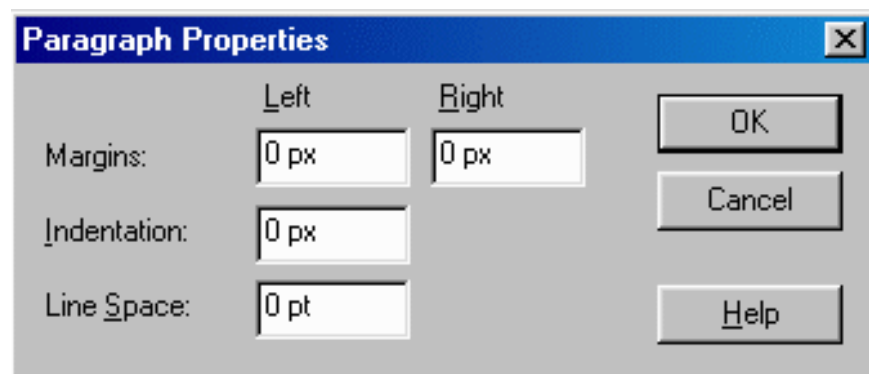
Alignment

You can change the alignment of your text by selecting from Left, Center, Right, and Justify.



Paragraph

{Click} the Paragraph modifier to change the behavior of Paragraph Text boxes. In the Paragraph Properties dialog-box that pops up, you can change the settings for Margins, Indentation, and Line Spacing.



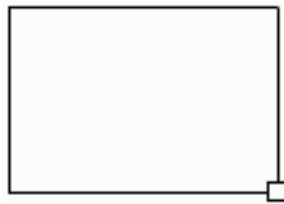
Text Field

Text Fields permit users to input or change existing text. In addition, text fields can be used to send information to servers or retrieve information from servers, similar to form fields in HTML. Creating and using Text Fields is covered in a later chapter. At this point, it's important to understand, that like regular text boxes, Text Fields can be expanding Text boxes, or Paragraph Text boxes. However, the process of creating them is reversed. To create an expanding Text box, with the Text Field modifier enabled, you {Click}, hold, and drag, then {Release} the mouse button, unlike regular text where you simply {Click}. To create Paragraph Text boxes, you simply {Click}, unlike regular text, where you {Click}, hold, drag, and {Release}.

Text Field boxes are easily recognizable because the hollow circle (expanding text box) or square (Paragraph Text box) is located at the bottom right rather than the top right.



Expanding Text Field box.



Paragraph Text Field Box.

Modifying Text

You can modify text in two ways. You can select the text box with the Arrow Tool and then *{Click}* the Text Tool. Use the modifiers to change the properties of the text.



A text box selected with the Arrow Tool.



The Font changed to Comic Sans from Arial Black.

You can also select the Text Tool and highlight the text inside the box by *{Clicking}* and dragging over the text. You can input different text and modify text properties.



Text highlighted.



“text” changed to “test” and font changed to Arial Narrow.

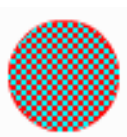
Experiment with the Text Tool. Use both types of text boxes. Modify existing text by changing size, color, font, etc. When you’re finished experimenting, select Edit/Select All from the Menu and press the [DELETE] key to clear the Stage.

The Overlay Level

You may have noticed how selected text does not look the same as selected shapes. Selected text has a selection box around it, while selected shapes display a checkered pattern.



Selected Text.



Selected Shape.

Recall that all shapes are created on the Shape Level (officially called the Stage Level). The Drawing Tools are only active on the Shape Level. If you try to change the color of text with the Paint Bucket Tool, or add an outline to text with the Ink Bottle Tool, you’ll be unsuccessful. Why? Because text is automatically created at the Overlay Level. To modify text on a shape level, such as extending the top part of the letter “t”, you have to bring the text down from the Overlay Level to the Shape Level. To do so, select the text with the Arrow Tool and select Modify/Break Apart from the Menu Bar.



Text on the Overlay
Level.



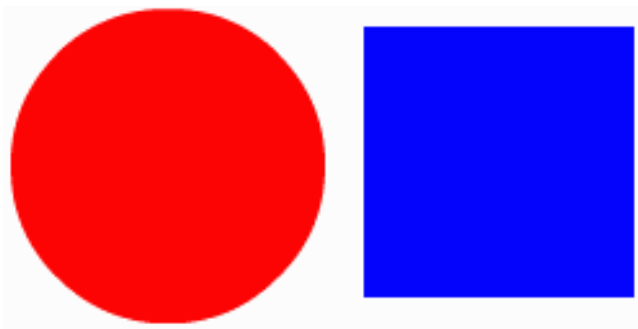
Text on the Shape
Level.

When you “Break Apart” text, you place it on the Shape Level. You can no longer edit the text as you can on the Overlay Level (by changing the content). The text, technically, is no longer text, but a collection of shapes. Now you can use the Drawing Tools to modify the shape.

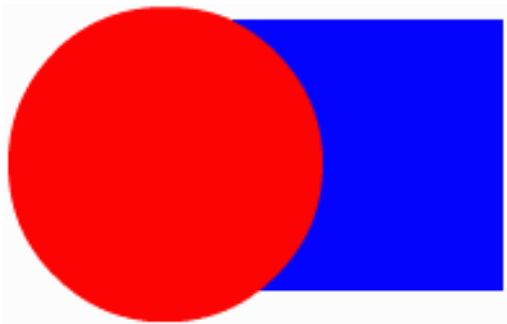


The modified shape.

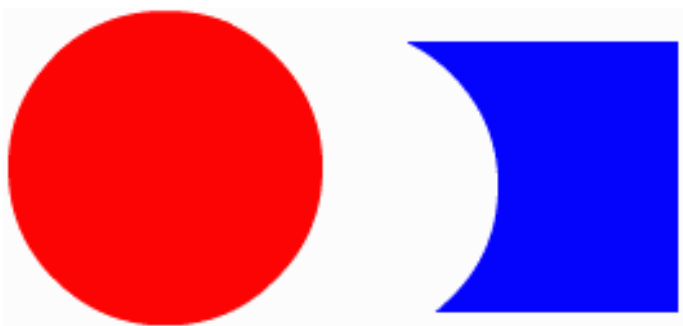
Think of the Shape Level as an area containing raw image (or shape) data. Each shape affects other shapes when they come in contact with each other.



Two shapes side by side on the Stage.



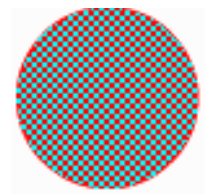
Two shapes overlapping each other.



Two shapes pulled apart. Notice how

the circular shape has cut into the rectangular shape.

When shapes are placed on the Overlay Level, one shape will not affect the other. To promote shapes on the Shape Level to the Overlay Level, you have to group the shape (or collection of shapes). Select the shape(s) and select Modify/Group from the Menu Bar. The selection indicator changes from a checkered pattern to a selection box.



Circular shape on the Shape Level.

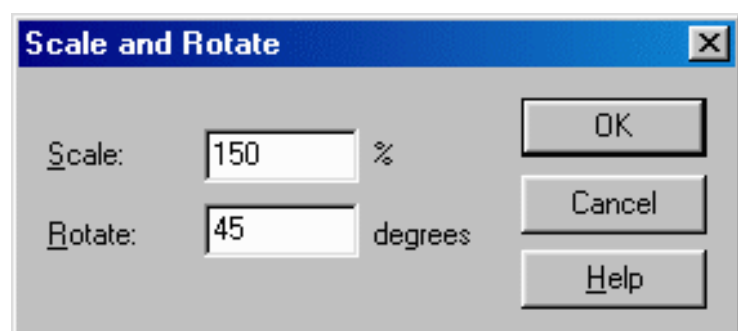


Same shape promoted to the Overlay Level.

To return a shape to the Shape Level from the Overlay Level, you select Modify/Break Apart, just as you do with text.

Scale and Rotate on the Overlay Level

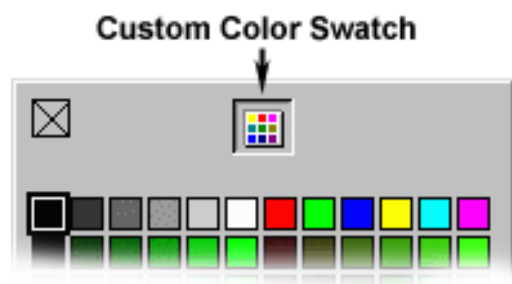
Note that the Scale and Rotation modifiers of the Arrow Tool function just as they do on the Shape Level. Simply select the shape with the Arrow Tool and use the Scale and Rotate modifiers accordingly. To scale and/or rotate shapes precisely, select Modify/Transform/Scale and Rotate from the Menu Bar. The Scale and Rotate dialog-box pops up.



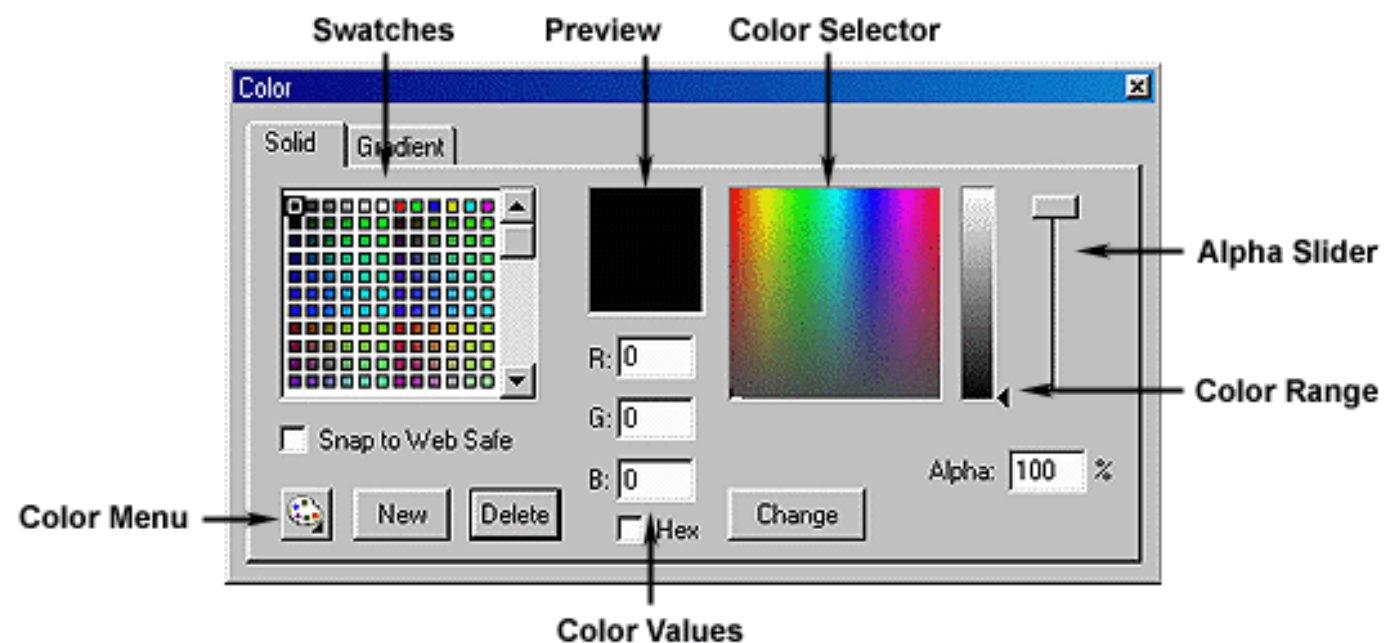
Input values into the Scale and Rotate fields. Note that inputting “100%” into the Scale field maintains the present size of the shape. Note also, that the values you input in the Scale and Rotate fields remain until you change them. This means that the next time you use the Scale and Rotate function, the values you used for the previous shape still remain. Make sure that you are only rotating a shape, the Scale value is “100%”.

Custom Colors and Gradients

You may need to create custom colors for your shapes, and you will certainly need to create custom gradient colors for your shapes. To create solid colors, {Click} the Custom Color swatch at the top of the Color Palette.



The Color dialog-box pops up.



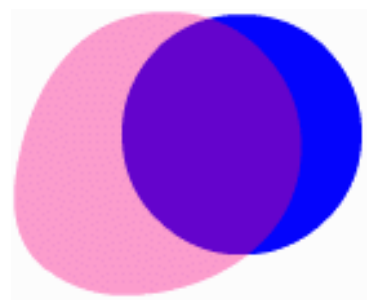
To create a new solid color, make sure the Solid tab is selected by {Clicking} on it. We don't recommend modifying existing colors. Select a color from the standard swatches from the Swatches area, to serve as the basis for your new color and {Click} the "New" button to add a new color swatch to the Color Palette. A new color swatch is added to the bottom of the swatches, and the color is displayed in the Preview swatch.

You can create a new color in a number of ways. You can {Click} over a color in the Color Selector. Check "Snap to Web Safe" to ensure the color you create conforms to web safe palette, although it will already be present in the standard swatches.

Use the Color Range control to cycle through the entire range of the color you've created. Basically, the color is lightened at the top of the range and darkened at the bottom.

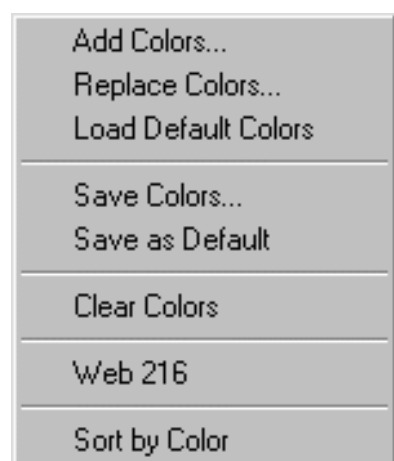
You can also input color values manually in either RGB values (Red, Green, and Blue), or Hexadecimal values. Check the "Hex" box to input Hexadecimal values

To add transparency to your color, move the Alpha Slider up and down to increase or decrease transparency. Shapes behind a color with transparency will show through the color with transparency, but only on the Overlay Level or higher (we cover the other levels later).



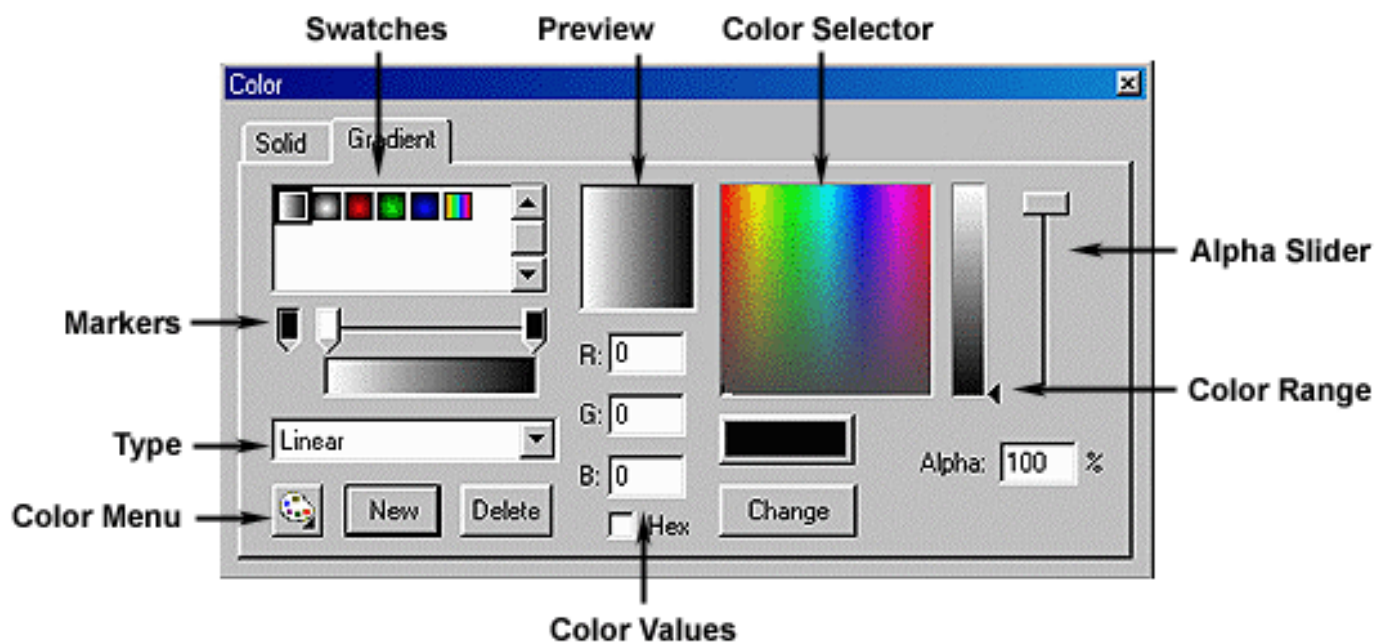
The egg shaped shape's color is slightly transparent, showing through the color of the other shape.

Use the Color Menu to load, save, and restore colors. We'll cover managing custom colors in a later chapter. For now, we recommend keeping with the web safe palette.



Custom Gradients

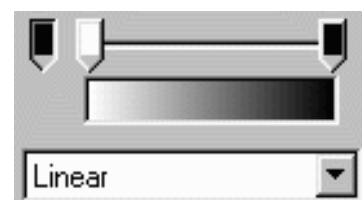
To create custom Gradients, *{Click}* on the Gradient tab of the Color dialog-box.



To create a custom gradient, select one of the preset Gradients on which you wish to base the new gradient and *{Click}* New. A new swatch is created. You create custom gradients by assigning colors to the Gradient Markers and by designating the Gradient Type: Linear or Radial. To add additional markers, simply *{Click}*, hold, and drag Markers over top the Marker Range. To remove markers, drag them off the Gradient bar. In addition to selecting colors from the Color Selector, you can also *{Click}* the color swatch below the Selector and select from the standard 216 web-safe colors.

Linear Gradients

Linear gradients are blends of colors that follow a straight path, from the color at the beginning of the gradient to the color at the end.



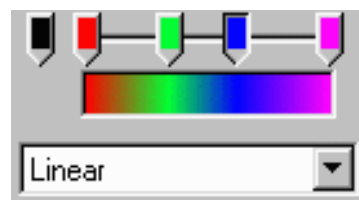
A Linear Gradient.



A rectangular shaped filled with a Linear Gradient.

In the examples above, there is a transition from white to black. The gradient begins with a white color and ends in black. Various shades of gray occur between.

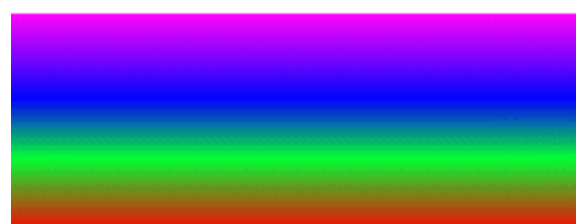
Gradients are not limited to two colors. Linear gradients with multiple colors produce a rainbow effect.



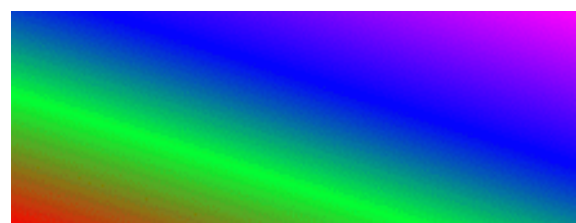
When using Linear gradients as fills, you can control the orientation of the fill, whether it is horizontal, vertical, or diagonal. Using the Paint Bucket, simply *{Click}*, hold, drag, and *{Release}*. To create a horizontal fill, you must drag the Paint Bucket vertically (up or down). To create a vertical fill, you must drag the Paint Bucket horizontally (left to right or right to left). Create a diagonal fill by dragging diagonally.



A horizontal gradient fill.



A vertical gradient fill.

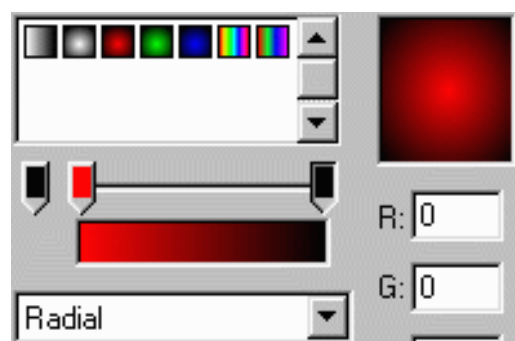


A diagonal gradient fill.

When you have completed creating your Linear gradient, close the Color dialog-box. A new swatch with your new gradient is added to the Color Palette.

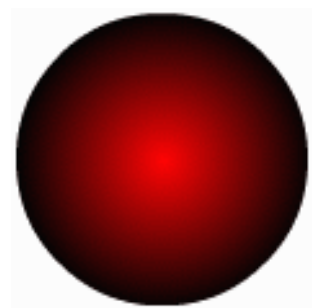
Radial Gradients

Radial gradients are blends of color from a center point outward. In Flash, the center point of a radial gradient is the first marker on the Gradient bar. The last marker represents the final color of the gradient.



A radial gradient.

You create Radial gradients just as you create Linear gradients, by dragging markers on to the Gradient bar and assigning particular colors to each marker. When filling a shape with a Radial gradient, where you *Click* the Paint Bucket is where the center point is placed.



A radial gradient. The center point is in the center of the circular shape.



A radial gradient. The center point is the top right of the circular shape.

Experiment with the Colors. Create custom gradients and fill shapes with them. When you're finished experimenting, select Edit/Select All from the Menu and press the [DELETE] key to clear the Stage.

Review Questions

1. How do you add a dotted outline with a thickness of 4 to a shape without an outline?
2. Why don't the drawing tools function on text?
3. What's the difference between a linear gradient and a radial gradient?
4. How do you smooth out an outline?

Summary

As a result of this chapter, you should be able to

- Use each of Flash's Selection Tools
- Use Selection Tool Modifiers
- Understand Text in Flash
- Understand Gradient fills and how to create new Gradients

CHAPTER 5

Flash Fundamentals: Symbols



To animate shapes, you have to make your shapes into Symbols. This chapter details how Symbols are created, how they work, how they are organized, and how they are edited.

OBJECTIVES

Upon completing this section, you should be able to:

1. Convert Shapes into Symbols
2. Use and understand the Library
3. Understand the Symbol and Instance Relationship
4. Modify Symbols
5. Modify Instances of Symbols

Symbols and Instances

Symbols are a third image level in Flash. Thus far, we've reviewed the Shape Level and the Overlay Level. At the Shape Level, each of the Drawing Tools is active, and each shape can affect the other. At the Overlay Level, the Drawing Tools do not function, but shapes at this level can be scaled, rotated, and moved on the Stage. Shapes on the Shape Level and the Overlay Level can be made into Symbols. When you convert a shape to a Symbol, the Symbol is placed in Flash's Library where it acts as the master reference for each Instance of the Symbol. What is an Instance? An Instance is a copy of the master Symbol. Rather than using the Symbol on the Stage, we use and modify the Instance of the Symbol on the Stage. Instances can be scaled, rotated, tinted, brightened, made transparent, and have colors changed either instantly or over time (in animation).

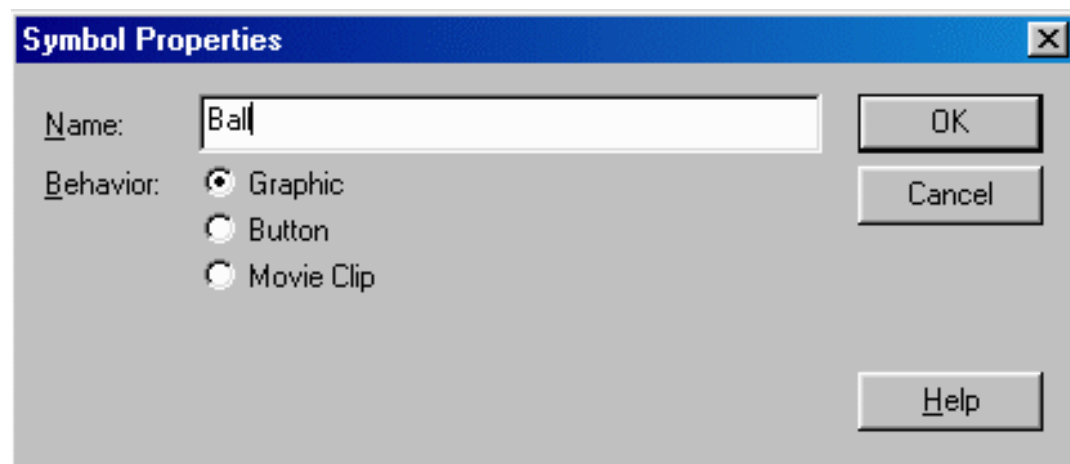
Creating a Static Symbol

To create a static Symbol (a symbol that is not animated), select an existing shape.



Select a shape you wish to convert to a Symbol.

Select Insert/Convert to Symbol from the Menu Bar and press [F8]. The Symbol Properties dialog-box pops up.



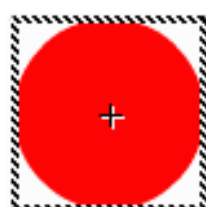
The Symbol Properties dialog-box consists of a Name field and three options for the type of Symbol you are creating under Behavior. In the Name field, enter a name that best describes the Symbol.

For Behavior, check “Graphic”. Graphic designates a static or still Symbol (a Symbol without animation or motion). We’ll cover both the Movie Clip (an animated Symbol), and Button (an interactive Symbol) Behaviors later in the manual. *{Click}* OK.

Now the shape is converted into a Symbol. Like a shape on the Overlay Level, a selection box surrounds a selected Symbol. However, there is one important difference. When a shape is a Symbol, a crosshair appears in the center of the selection box.



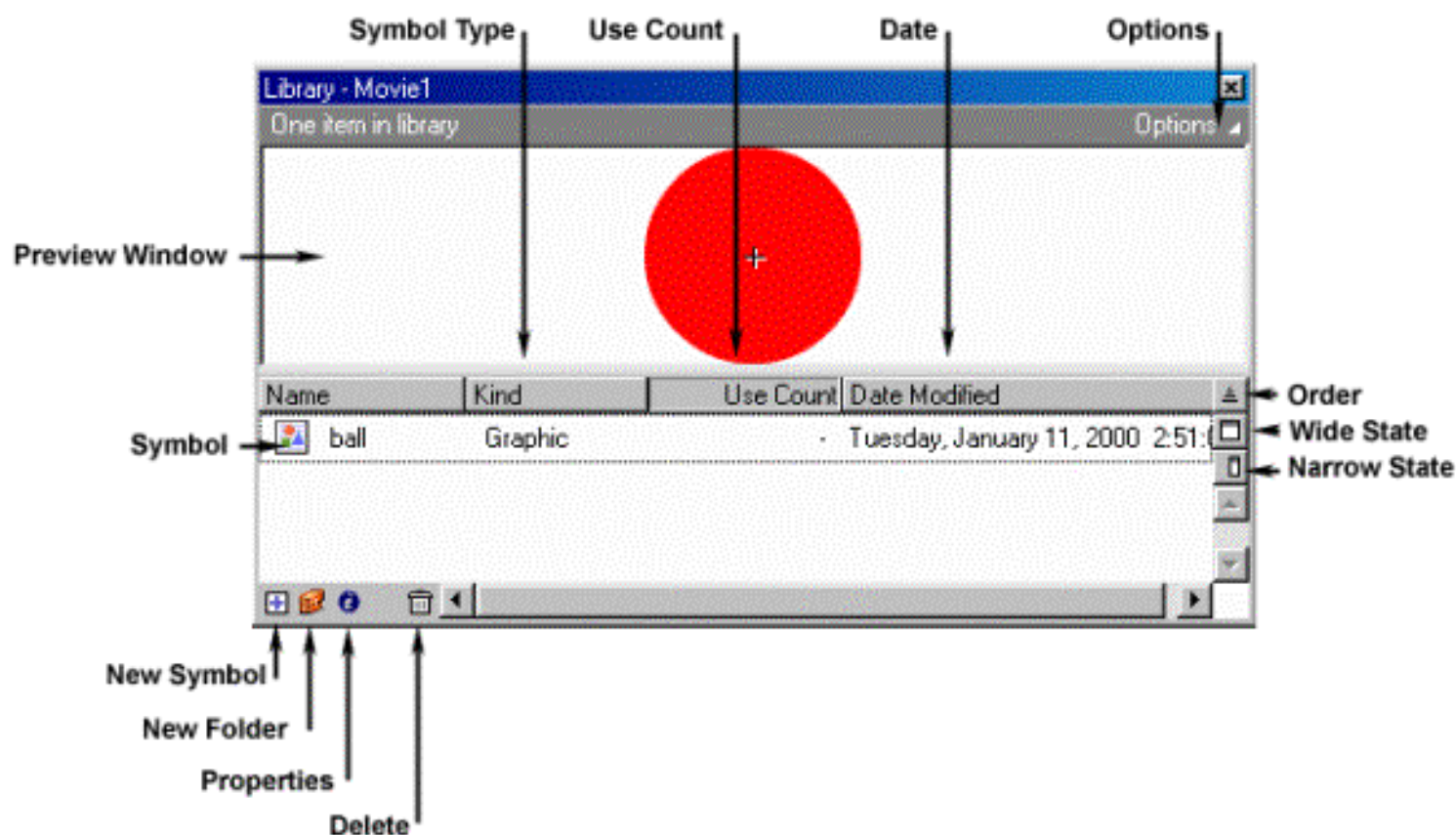
A selected shape on the Overlay Level.



A selected shape as a Symbol. Note the crosshair in the center of the selection box.

The Library

Symbols are automatically placed into the Library. The Library is used to store and organize Symbols. To open the Library, select Window/Library. The Library pops-up.



The Library Window. Use the Library to store and manage Symbols.

A list of all the Symbols in the current movie is displayed below the Preview Window. The icon beside the name of the Symbol designates the type of Symbol (the Behavior). The Behavior is also listed under “Kind”. The Symbol itself is displayed in the Preview Window, and must be selected by *{Clicking}* the Symbol to view it.

 Graphic Behavior Icon

 Movie Clip Behavior Icon

 Button Behavior Icon

To sort Symbols, click on the “Kind” tab to sort by Symbol Type, the “Use Count” tab to sort by use, or the “Date Modified” tab to sort by date. You can also select a Symbol and *{Click}* on the “Order” arrow to move it around the list.

{Click} the “Wide State” or “Narrow State” buttons to modify the width of the Library Window.

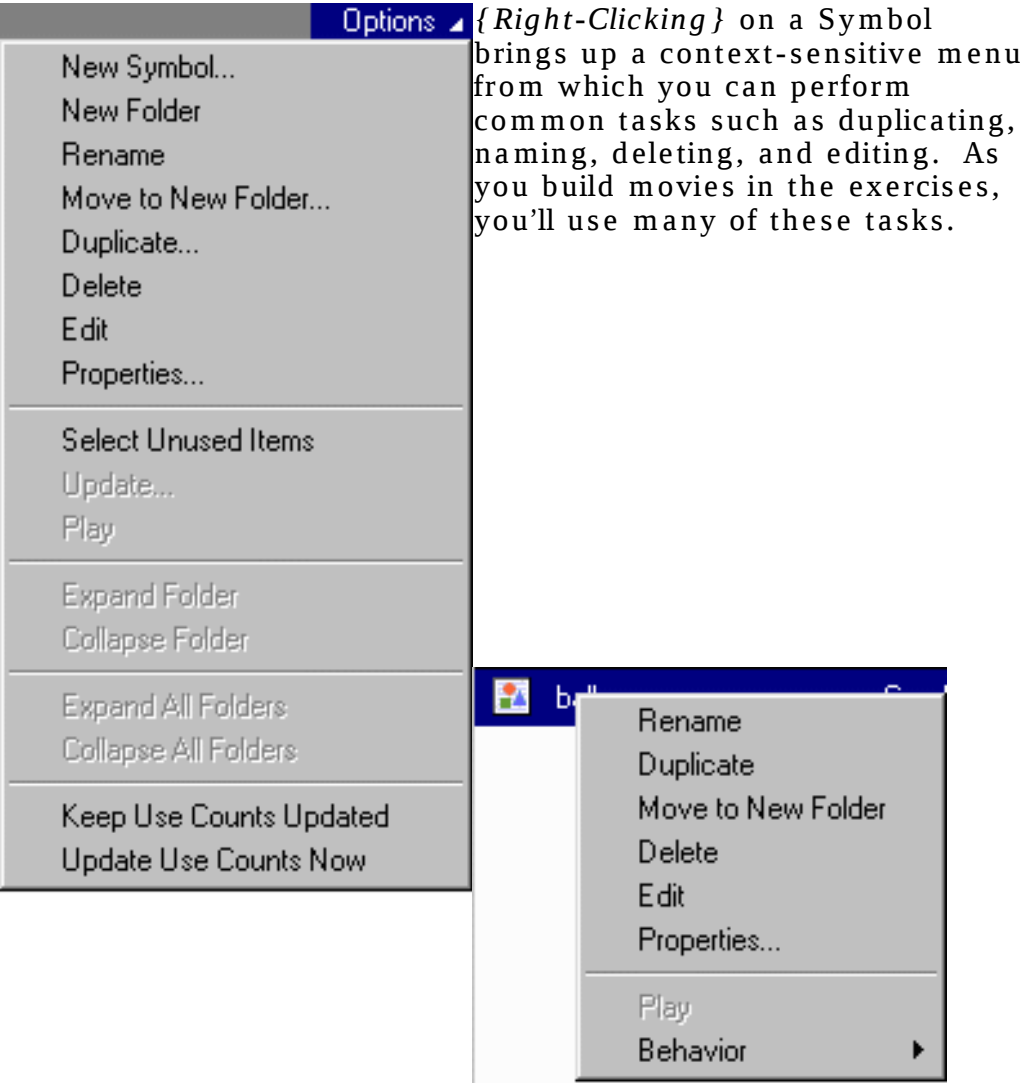
As you produce Flash movies, you’ll find yourself using a lot of Symbols. Symbol counts, depending on the movie, can run into the hundreds. Fortunately, the Library permits you to create folders so you can organize your Symbols. Create new folders by *{Clicking}* on the “New Folder” icon . As you do the exercises, you’ll be creating Folders in the Library.

{Click} the “Properties” icon  to change the name of the

Symbol, change the Behavior, and edit the Symbol. We cover Symbol editing later in the chapter.

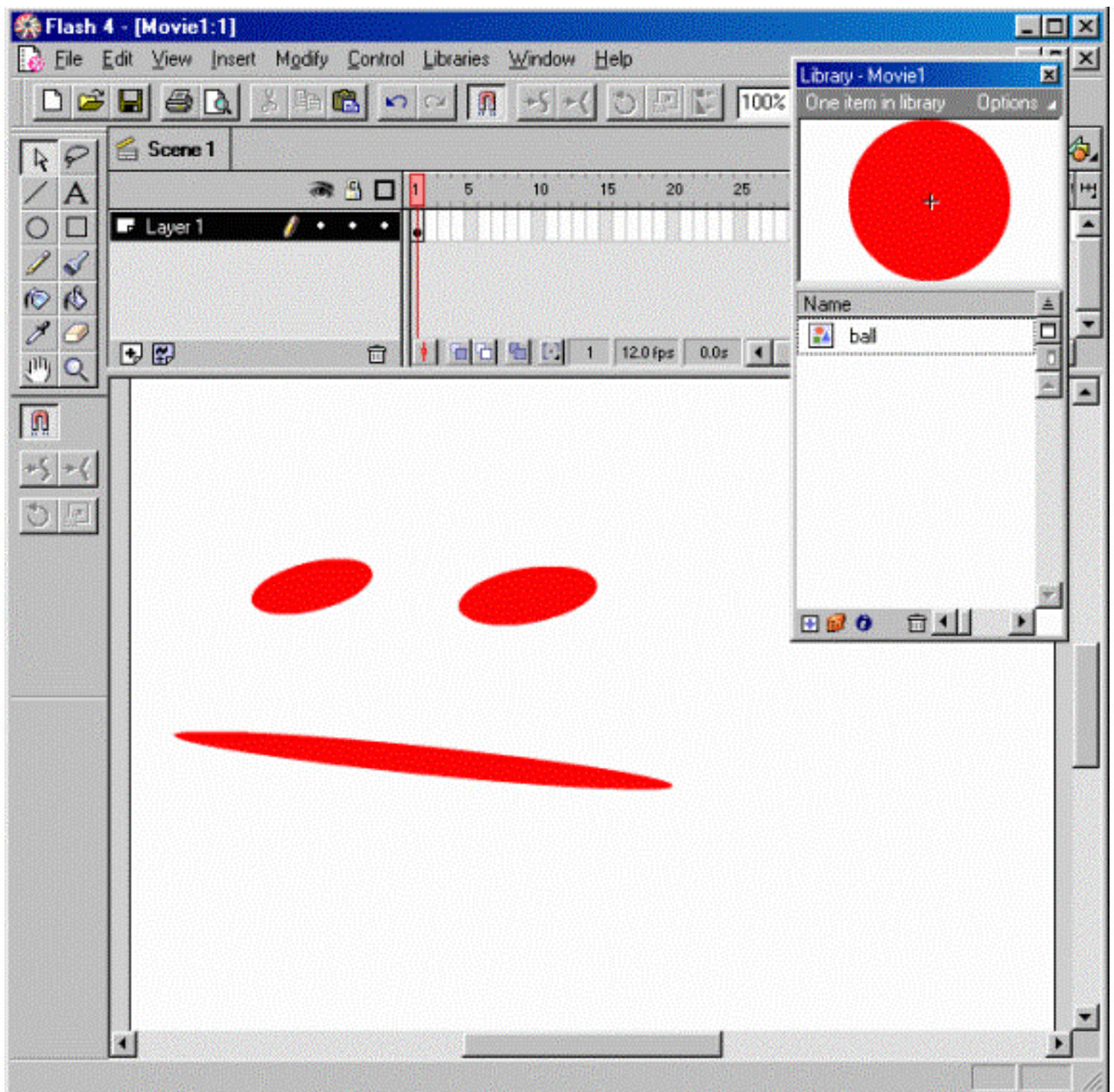
To delete a Symbol, select it and {Click} the Delete icon .

Many of the same functions provided by the various icons in the Library Window are also accessible in the Options menu. {Click} the Options drop-down area and select the item you desire.



Using Instances

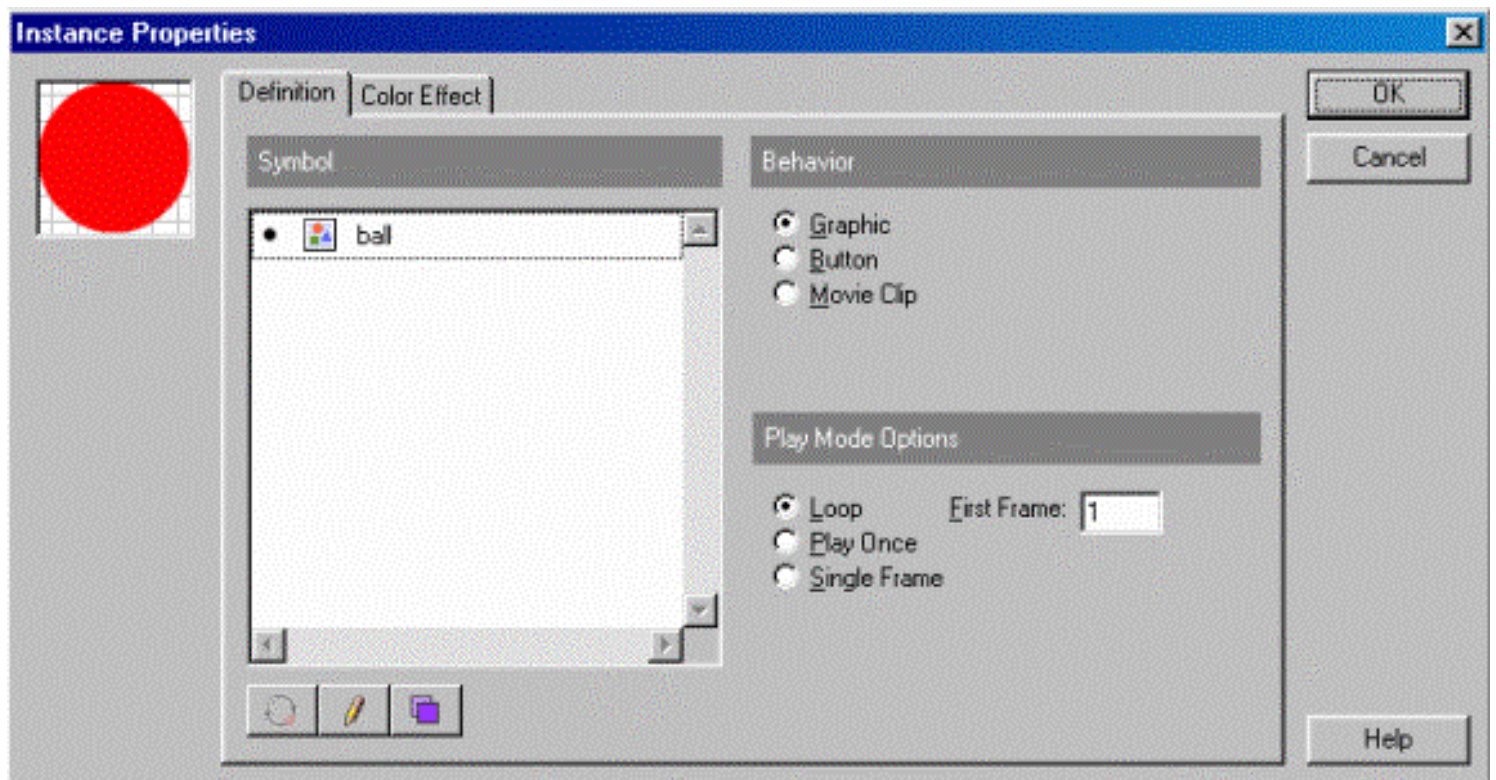
To use an Instance of a Symbol, select the Symbol in the Library and drag it onto the Stage. Keep in mind that the actual Symbol remains in the Library. The shape on the Stage is an Instance (or copy) of the Symbol. You cannot use the Drawing Tools on the Instance; however, you can Scale and Rotate the Instance. Scaling and Rotating the Instance has no effect on the Symbol. This is the chief advantage of Symbols and Instances. You modify the Instance of the Symbol while keeping the master reference unchanged for future use. In addition, you can use multiple Instances based on one Symbol. For instance, if we need three red circles on the Stage, we can drag three Instances of the “ball” Symbol onto the Stage. We can Scale and Rotate each Instance as we please. Flash uses the Symbol as a basis for the Instance and records the changes you’ve made to the Instance on the Stage. This is an efficient use of shapes, both in production and in actual file size. If we had to create three separate circular shapes, it would take that much longer and increase the file size of the movie.



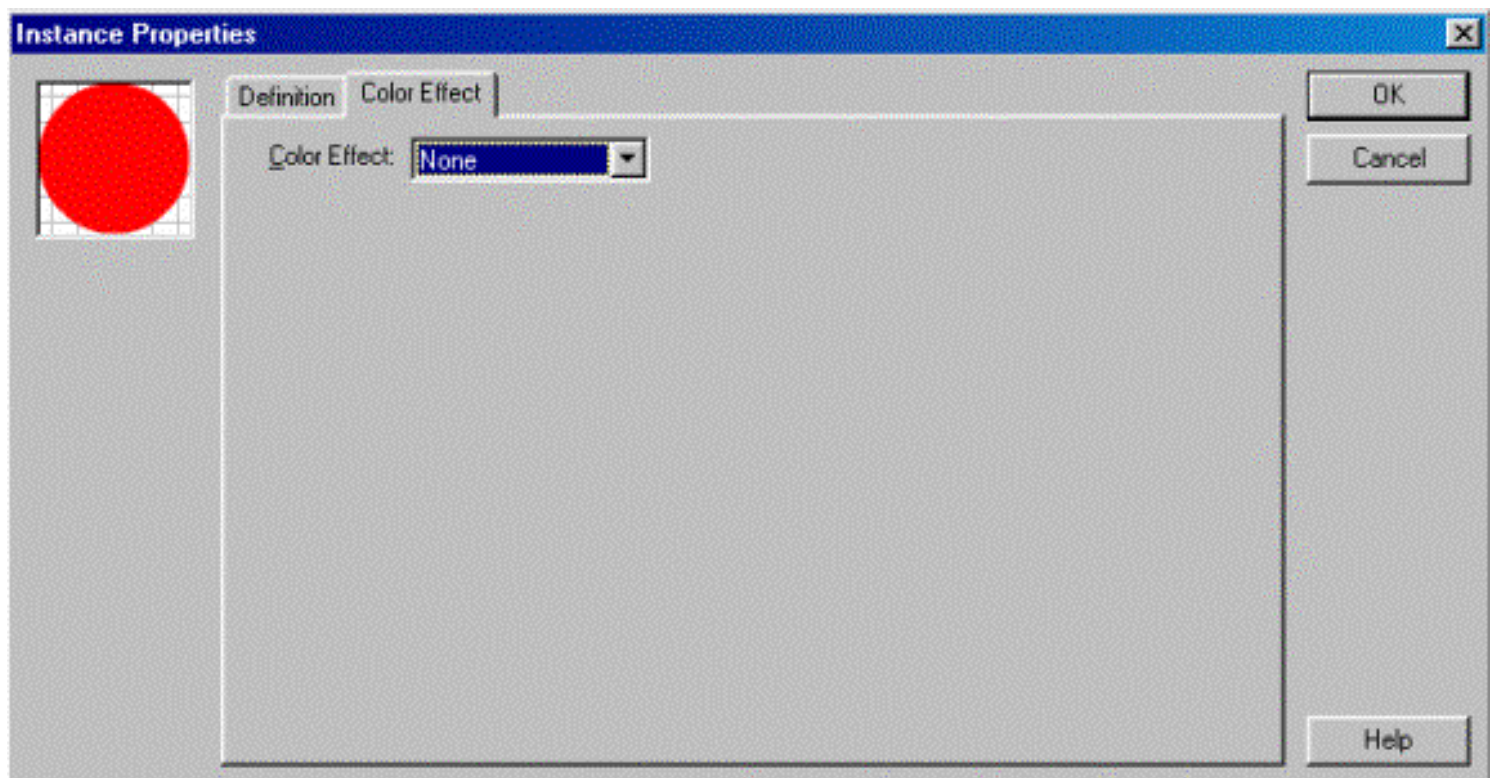
This movie has one Symbol (ball). There are three (3) Instances of the Symbol on the Stage. Each Instance has been positioned, scaled, and rotated.

Instance Properties

In addition to Scaling and Rotating Instances, you can also modify Instance Properties. To modify Instance Properties, select the Instance and select Modify/Instance from the Menu Bar or {Right-Click} the Instance and select Properties from the context-sensitive menu. Alternately, you can select the Instance and press [CTRL] + I. The Instance Properties dialog-box pops up.



We'll be covering the Definition aspect of Instance Properties in a later chapter, for now, *{Click}* the Color Effect tab.

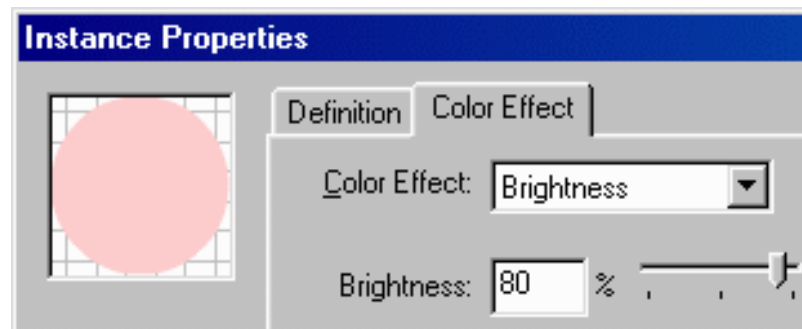


{Click} the Color Effect drop-down menu to select from Brightness, Tint, Alpha, and Special.

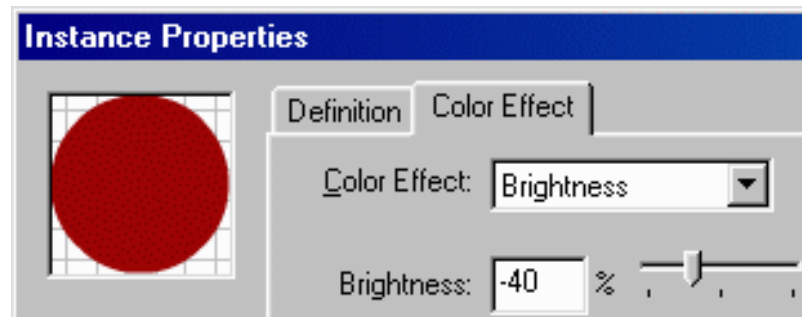


Brightness

Modify the Brightness of an Instance by increasing or decreasing the Brightness in terms of percent. A preview of the effect is displayed beside the Brightness control.



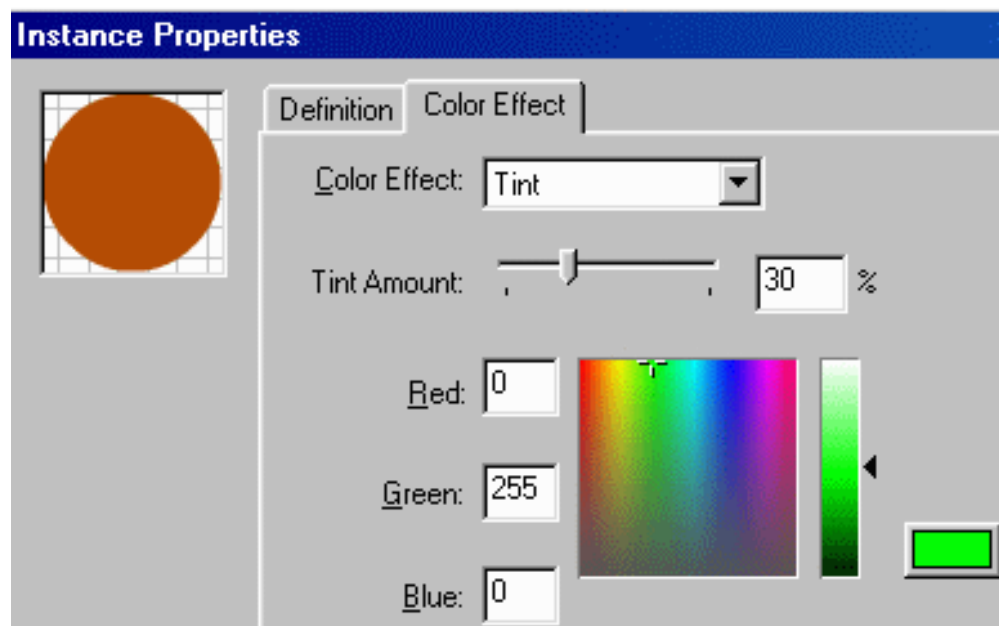
The Instance brightened by 80%.



The Instance darkened by -40%.

Tint

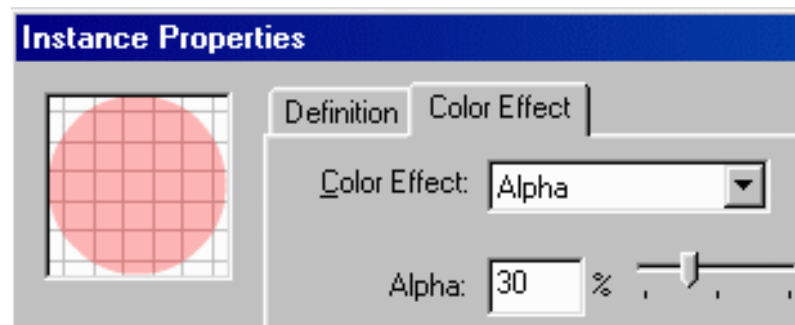
Tint (modify an existing color by adding another one to it) an Instance using the Tint controls. Create a color by using the Color Selector, inputting RGB values, or selecting a color from the Color Palette by {Clicking} on the color swatch. Use the Tint Amount slider to tint the existing Instance or input a value into the Tint field.



The Instance is tinted with green at a level of 30%.

Alpha

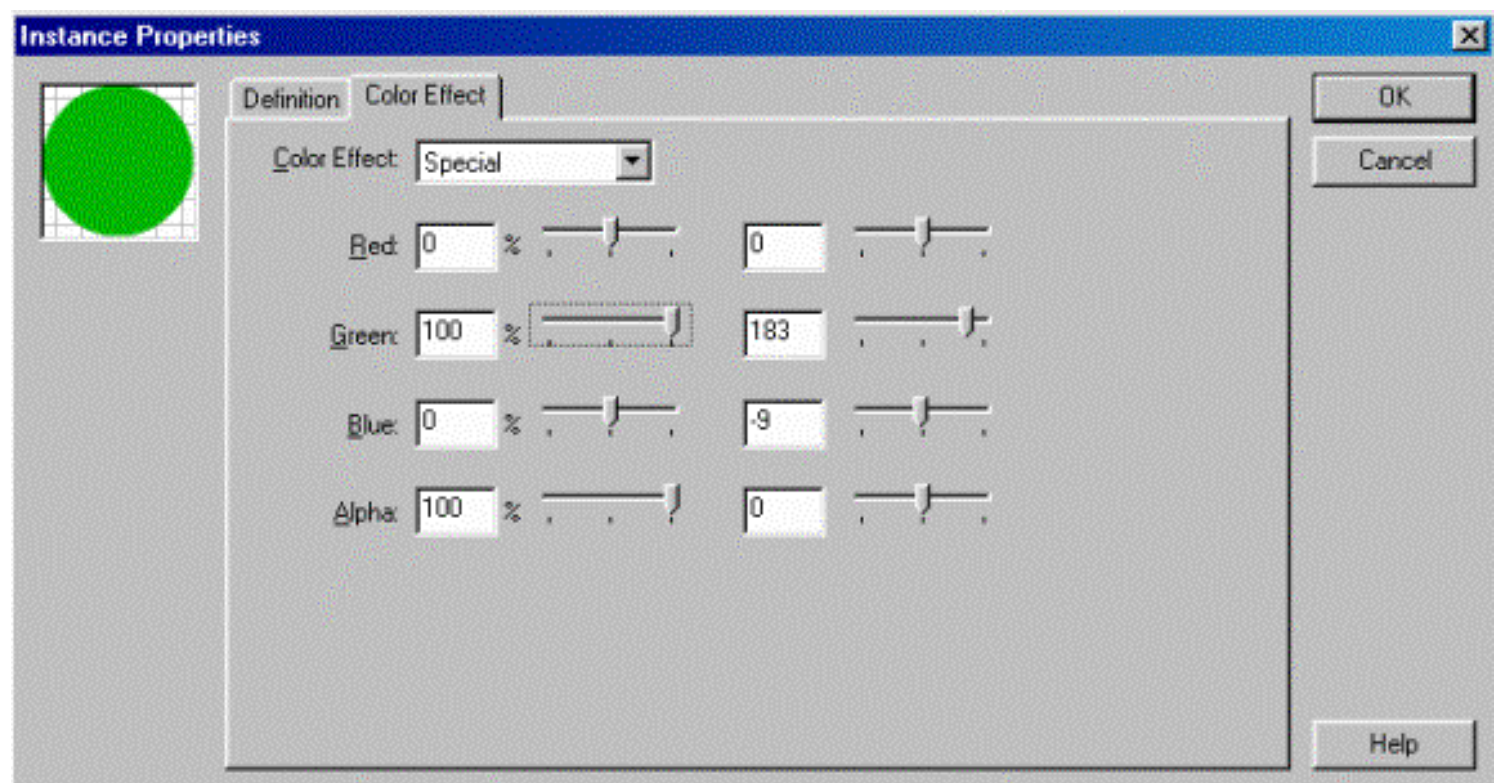
Add transparency to Instances with the Alpha controls. Use the Alpha slider to increase or decrease transparency. 100% is fully opaque, 0% is totally transparent.



The Instance made 30% transparent.

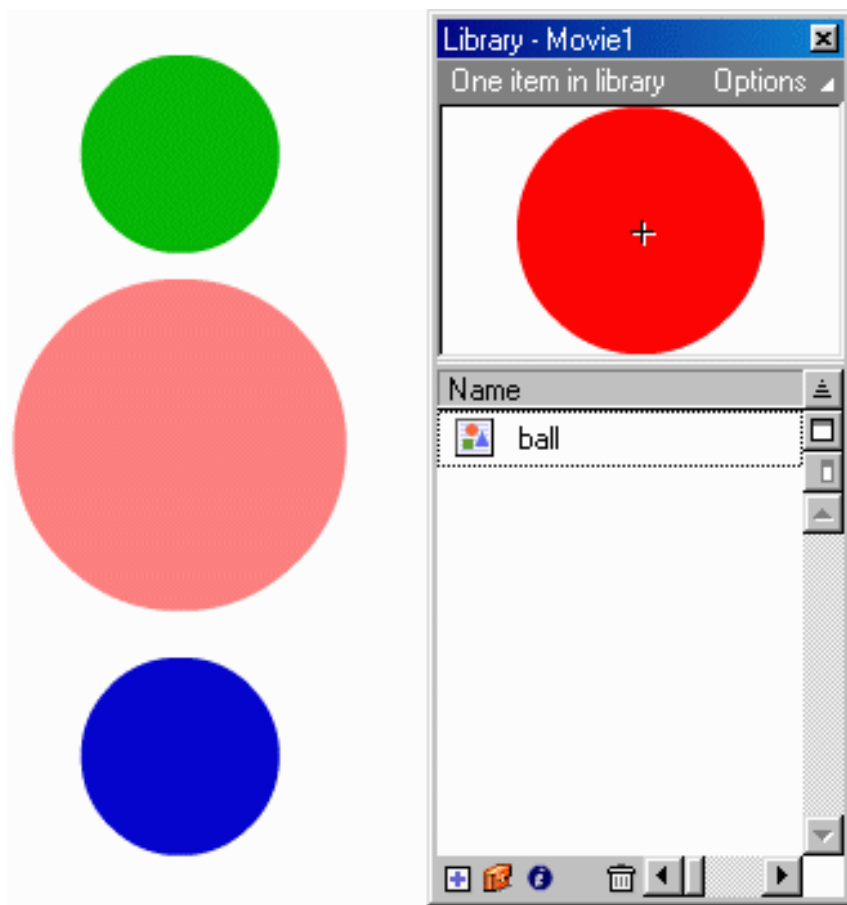
Special

Change the color of shapes using the Special controls. Use the color sliders to mix various levels of Red, Green, and Blue. Use the Alpha slider to add transparency.



The Instance color is changed to green from red.

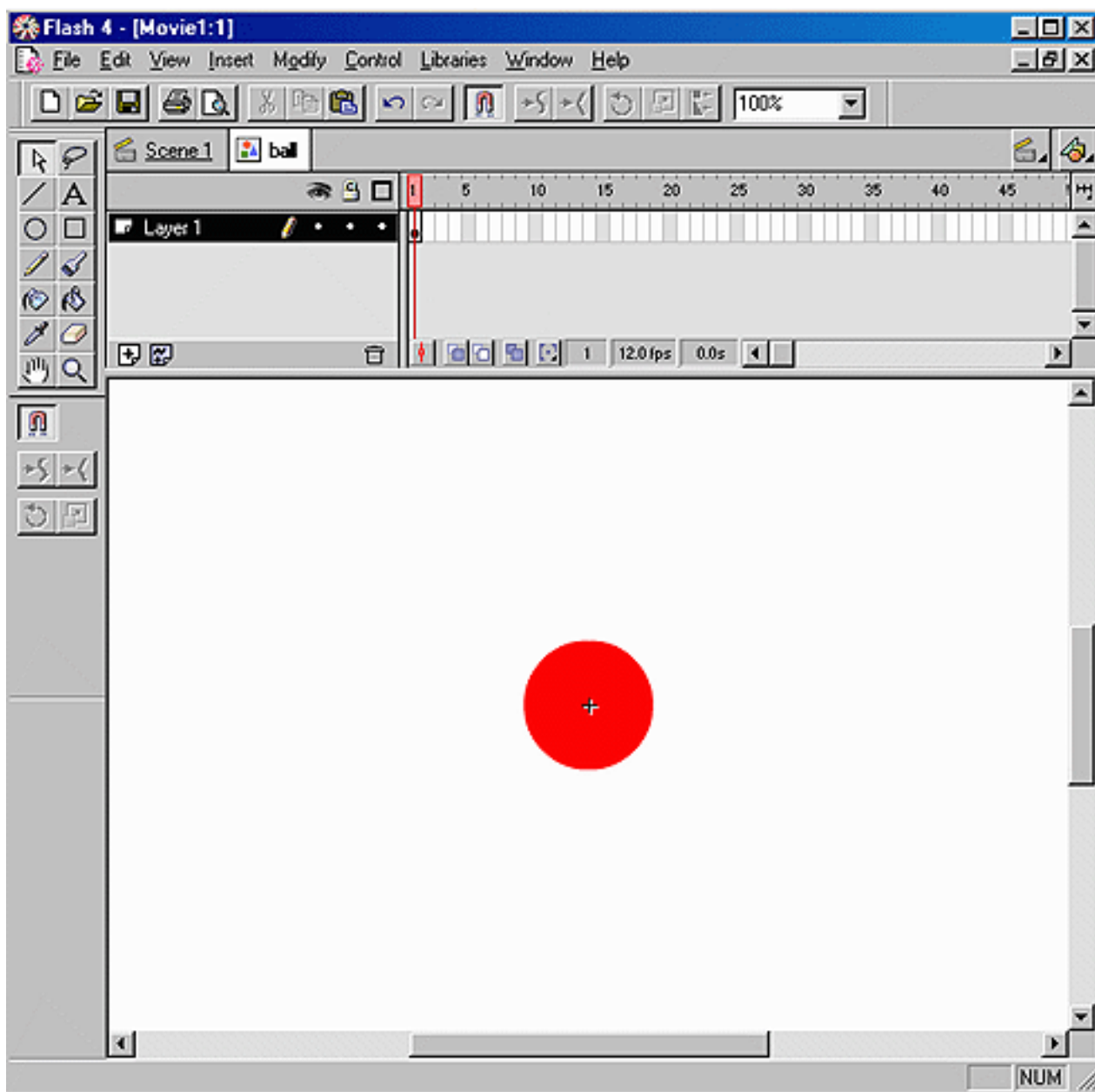
With the ability to change Instance Properties, you can make Symbols made up of primary shapes and modify Instances when required. Each time you need to use a circle, for example, you need only make one Symbol of a circular shape and modify Instances of that Symbol throughout your movie.



Each Instance is based on one Symbol.

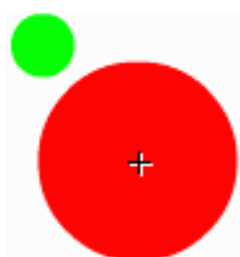
Editing Symbols

Another great advantage of Symbols and Instances is the ability to change Instances throughout your movie by editing the Symbols upon which they are based. To edit a Symbol, {*Right-Click*} the Symbol (not the Preview Window) in the Library and select “Edit” from the context-sensitive menu. You are taken out of the main Stage area and placed into the Symbol’s Stage. Each Symbol has its own Stage upon which you modify the shape(s) that make up the Symbol. There are two visual indicators to show you are not on the main Stage but on a Symbol Stage. One, a crosshair appears in the center of the stage, just as there is a crosshair in the center of an Instance’s selection box. Two, a Symbol Tab appears beside the Scene Tab above the Timeline.



A cross hair appears in the center of Stage, indicating we are actually in the Symbol's stage rather than Flash's main Stage. A Symbol Tab, with the name of the Symbol, appears beside the Scene Tab above the Timeline, also indicating we are on a Symbol Stage.

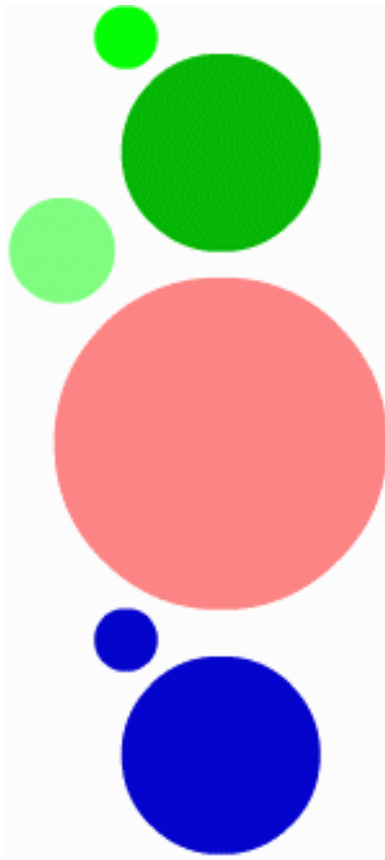
Because you are essentially inside the Symbol, you can modify the shape as you normally would on the main Stage. Shapes within Symbols can be on the Shape Level, the Overlay Level, and can even be other Symbols (in which case you can only modify Scale, Rotation, and Instance Properties just as on the main Stage). We're going to add another circular shape above and to the left of the existing shape.



We've added a second circular shape to the

existing Symbol.

Once you've completed editing a Symbol, *{Click}* the Scene Tab above the Timeline to return to the main Timeline. Because we modified the original Symbol, each Instance is modified on the Stage.



The Instances now have the additional circular shape.

It's very important to understand that Flash has a main, or a master Stage, and each Symbol also has its own Stage. Because each Stage is tied to a Timeline, we refer to each type of Stage as either the master Timeline or a specific Instance Timeline. Shapes, and the various image Levels (Shape, Overlay, Symbol) work precisely the same way on each type of Stage. Understanding the Symbol and Instance relationship is also essential to using Flash effectively. With careful planning, you can ensure that your Flash files remain small in terms of file size, and that changes can be made movie-wide simply by changing the Symbol.

Review Questions

1. How do you convert a shape to a Symbol?
2. How do you edit a Symbol?
3. Describe the Symbol/Instance relationship?
4. When an Instance is on the Stage, how can you modify it?
5. When a shape is selected, how can you tell if it is on the Shape Level, the Overlay Level, or the Symbol Level?
6. What is the function of the Library in Flash?

Summary

As a result of this chapter, you should be able to

- Convert Shapes into Symbols
- Use and understand the Library
- Understand the Symbol and Instance Relationship
- Modify Symbols
- Modify Instances of Symbols

CHAPTER 6

Flash Fundamentals: Animation Techniques



Animation is one of the core functions of Flash. Animation, however, isn't simply moving from point A to point B. There are many factors to consider in creating effective and compelling animations. This chapter reviews animation basics, and introduces you to Flash's Timeline by creating the most traditional of animations, a frame-by-frame animation.

OBJECTIVES

Upon completing this section, you should be able to:

1. Understand Animation basics
2. Use the Timeline
3. Create Frame-by-Frame Animations

Animation Basics

Traditional animation involves a series of still images, usually painted or sketched, displayed in rapid sequence. Because the receptors in our retina require recovery time, we are unable to distinguish each individual image in the sequence. In effect, as the previous image is replaced by a new image the previous image is still burned on our retinas. Each image is blurred into the other, creating the illusion of motion. This is called Persistence of Vision. If we didn't experience retinal lag, we'd be able to distinguish each individual frame, destroying the illusion of motion.

With film, each image is called a frame. As a result, the rate at which each frame is displayed is measured in frames per second (fps). The movies you see at a theater display 24 frames per second. Videotapes and television display 30 frames per second. Film and video have fixed frame rates. With Flash, you can set the frame rate. Higher frame rates result in smoother animation, but also require more processing power.

Key Frames and In-between Frames

Animation is a laborious task. An animated movie, playing at 24 fps, at a total duration of 90 minutes, requires 129600 individual frames to be drawn and painted. Each of these frames, at least before the advent of computer animation, must be drawn by hand. To facilitate production, there are Key Frame animators and In-between Frame animators.

Key Frames

Key Frames are frames that act as reference points, usually illustrating a major moment in time. For example, if you were to create an animation of a runner practicing a 100-meter dash, you could have the following Key Frames:

Key Frame 1: Runner Standing Straight

Key Frame 2: Runner crouched in the blocks

Key Frame 3: Runner at full stride

Key Frame 4: Runner crossing the line

The Key Frame artist (or a team of Key Frame artists) is responsible for drawing each Key Frame. If this entire sequence is to last for 15 seconds (at a frame rate of 24fps), we'll obviously need more than four frames to animate it (360 frames in total). This is where the In-between Frame artists come in.

In-between Frames

The In-between artists are responsible for drawing the frames in between the Key Frames. For example, by examining the Runner standing straight, and then crouched in the blocks, the In-between artists know they have to draw the transition from a standing position to a crouching position. As a result, this process is called In-betweening, and in Flash is referred to as Tweening. Fortunately, in most cases, you act as the Key Frame artist while Flash acts as the In-between artist.

Motion

If you think about the real world, each object, whether it is inanimate (objects without life) or animate (objects with life), is subject to the physics of motion. Energy, gravity, momentum, and resistance are major factors in motion. As our runner accelerates, the energy he generates determines how quickly he gets out of the block and how quickly he gets up to speed. Running takes a lot of energy because gravity is pulling the runner's legs toward the ground and the air he is running through counters his momentum. As he passes over the finish line, the runner will want to stop before he hits the barrier. How quickly the runner comes to a complete stop depends on the speed he is running at as he passes the finish line and the amount of energy he can expound to reduce his momentum. Symbols in Flash are not subject to the laws of physics, but because people are used to seeing motion that is subject to physics, creating effective animation involves simulating or mimicking these laws. With computer animation, it's easy to have symbols move from point A to point B in constant motion, meaning the symbol is instantly at speed from the moment it begins to move to the moment it stops. Because people are used to natural motion (motion subject to physics) people respond well to animation that also follows the laws of physics, or at least gives an impression that physics are a factor. Constant motion is unnatural (there are exceptions), making the viewer less responsive and comfortable when visiting your web site. The degree to which you simulate physics in your animation will vary depending on the type of animation you develop, but even things as simple as animated buttons benefit greatly if animated in a realistic manner.

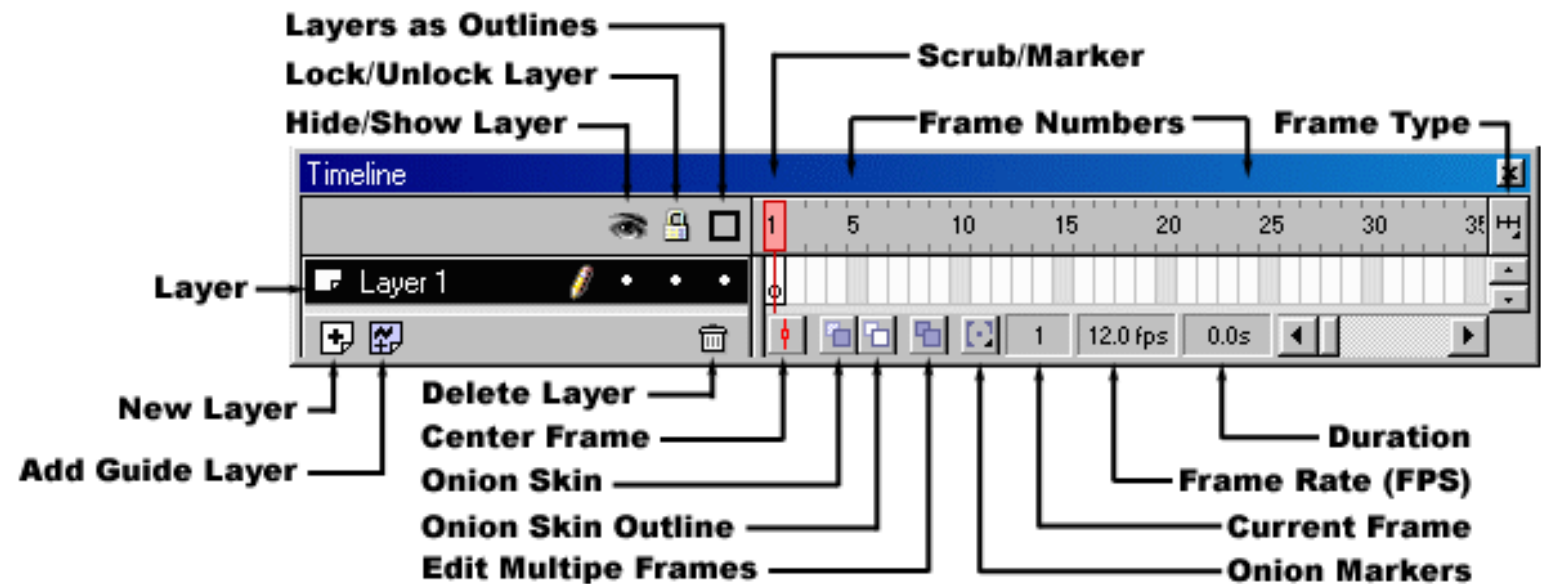
If you think of our runner once more, he will be racing against other runners. Each runner moves at a particular pace, from the

fastest to the slowest. When the winner crosses the finish line and breaks the tape, the tape also moves. It stretches, breaks, and then falls to the ground. Giving variable pace to different Symbols in your Flash animations also enhances the effect.

Open the Constant.fla file in the Chapter06 folder and examine the motion.

Open the In.fla file in the Chapter06 folder and examine the motion.

Open the Out.fla file in the Chapter06 folder and examine the motion.



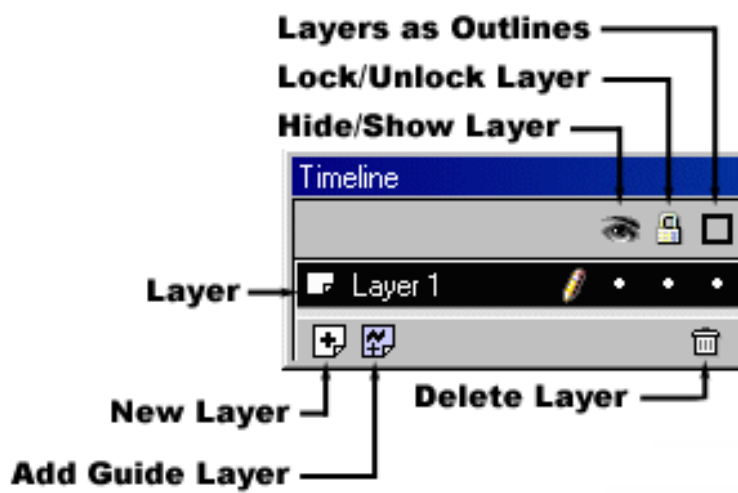
The Timeline

Animations are built with the Timeline. The Timeline is divided into Layers. Layers are similar to acetate or transparent paper where individual Symbols are placed and animated. Each Layer is stacked, one on top of the other, which, when combined, create an animation or image composed of many different elements. In Flash, only one Symbol per Layer can be animated.

The Timeline also contains controls to edit and manage animation, including onion skin controls, the ability to edit multiple frames, the ability to move to specific frames within animations, as well as customizing the display of Layers.

Layer Controls

By default, Flash displays one Layer labeled Layer 1. It's a good idea to label each Layer according to content. To label Layers, simply {Double-Click} the Layer label and input a new name.



New Layer

As you use Symbols to build animations, you'll need to add Layers for each Symbol you animate. To add new Layers, {Click} the New Layer button. A new Layer is added. You should label the Layer accordingly.

Guide Layer

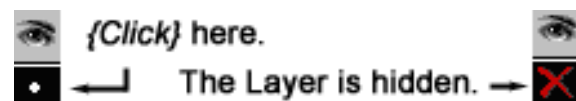
There are two specialized Layer types, Guide and Mask Layers. Both are used for specialized types of animation that are covered later. Briefly, when you use Guides, {Click} the Add Guide Layer button to add a Layer with content that acts as an animation guide.

Delete Layer

To remove a Layer from the Timeline, select the Layer and {Click} the Delete Layer icon.

Hide/Show Layer

As you build animations, you may want to hide the contents of a Layer or multiple Layers to assist in editing. Select the Layer you wish to hide and {Click} the dot in the Hide/Show Layer column. A red x appears indicating the Layer is hidden. Any content on that Layer is hidden on the Stage. {Click} the red x to make the Layer visible again.



Alternately, you can hide or show all Layers by {Clicking} on Hide/Show Layer icon  itself.

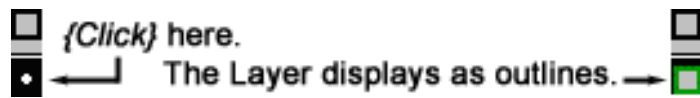
Lock/Unlock Layer

As you build animations, you may want to lock the contents of a Layer or Layers to prevent accidentally selecting and moving content. Select the Layer you wish to lock and {Click} the dot in the Lock/Unlock Layer column. A lock appears indicating the Layer is locked. You will not be able to move or edit the content of a locked Layer until you unlock the Layer. {Click} the red x to make the Layer visible again.

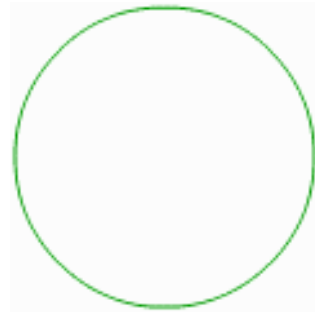


Outline Layer

As you build animations, you may set Symbols to 0% Alpha, which makes them invisible on the Stage. This can make it difficult to edit animations when content isn't visible. *{Clicking}* the dot in the Outline Layer column causes the content of the Layer to display as an outline, making Symbols set to 0% Alpha visible. The outline box is colored, indicating the Layer will display outlines.



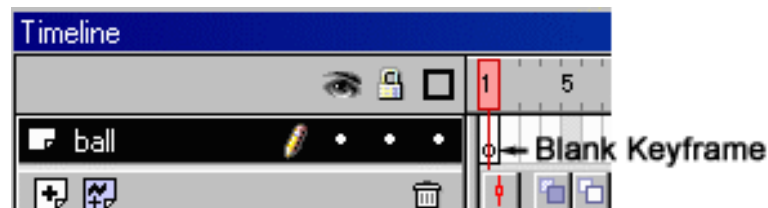
Symbol at 100 % Alpha.



Symbol at 0 % Alpha with Outline Layer selected.

Creating Animations with the Timeline

Once you have created the Symbols for your Flash movie, you use the Timeline to animate Instances of each Symbol. When you animate an Instance, you are limited to one Instance per Layer. The first thing you should do is create a new Layer for the Instance you want to animate. Think of each Layer as a roll of film, which you view through the Stage. By default, a blank key frame is placed in the first frame of the Layer. Each frame is separated by a line and a frame number appears above each frame every 5 frames.



If you wish to start your animation at the first frame, simply drag an Instance onto the Stage. The key frame changes from an empty circle to a filled circle, indicating that at that moment in time, there is an Instance on the Stage within this particular Layer.

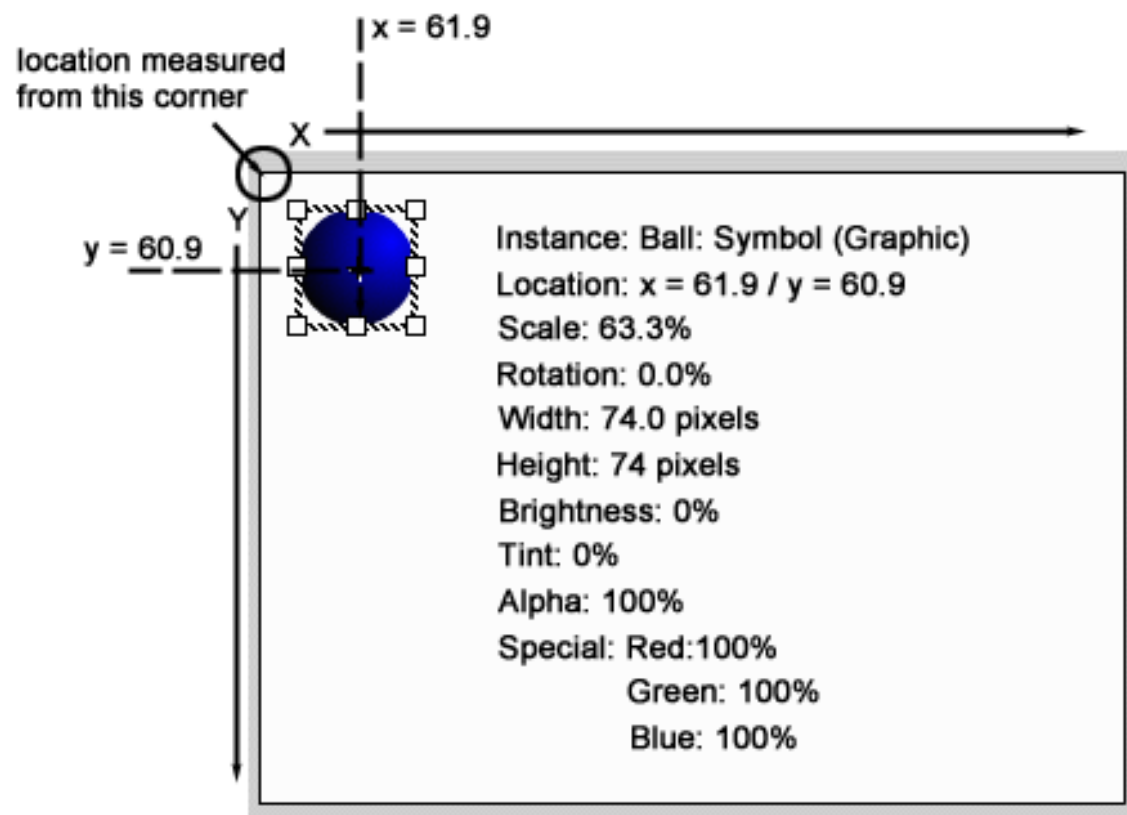


Key Frames in Flash

When you create a key frame in Flash, the key frame keeps track of the following properties:

- The Instance and the Symbol the Instance is based on
- The location of the Instance on the Stage (x and y coordinates, measured from the center point of the instance relative to the top, left corner of the Stage)
- The Scale of the Instance relative to the original Scale of the Symbol
- The Dimensions of the Instance in terms of Width and Height
- The Rotation value of the Instance
- Any Instance Properties that have been applied to the Instance such as Brightness, Tint, Alpha, and Special

For example, the following Instance, derived from a Symbol called Ball, would have the following information attached to the key frame it is placed in:

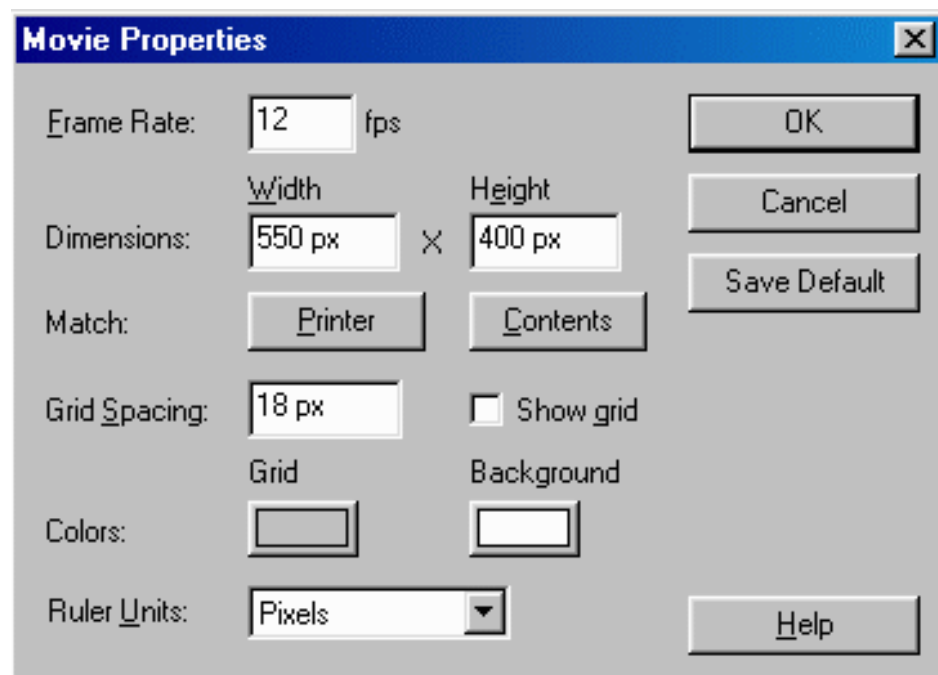


Flash uses this information to instruct the computer what to draw through each frame of the animation as well as calculating in-between frames.

At this point, depending on the type of animation you are creating, you can animate the Instance in a number of different ways. Our animation consists of a ball falling to the ground and bouncing up. The first thing we should do before beginning to build an animation is set up the Flash movie parameters, most particularly the frame rate (fps), Stage dimensions, and Stage color. Select Modify/Movie from the main menu. The Movie Properties dialog-box pops up.

Movie Properties

It is essential to set up the parameters of your Flash movie before you create your animations. By default, Flash movies are set on a Stage with a white background, at 550 pixels wide by 400 pixels high, at 12 frames per second.



Frame Rate

Set the frame rate (fps) to a minimum of 12. Keep in mind that the greater the frame rate, the greater the processing power required. You may find that 12 fps is too slow (the animation is not smooth). Try a range between 15 – 20 fps. As you create animations, you'll find a frame rate that complements your particular style. In regards to the Timeline, if you have set your frame rate at 15 fps, then each individual frame in each Layer is 1 / 15 of a second in time.

Dimensions

The default Stage dimensions of 550 pixels wide by 400 pixels high is intended for display at 640 x 480, the lowest common denominator in terms of screen resolution. However, many web sites are a mixture of Flash and traditional HTML, so you won't always develop movies that are intended to fill the entire browser window. You might want to develop a Flash based navigation system that is 200 pixels wide by 350 pixels high, or a banner that is 468 pixels wide by 60 pixels high. Depending on the type of movie you are creating, you will input appropriate dimensions.

Grid Spacing / Show Grid

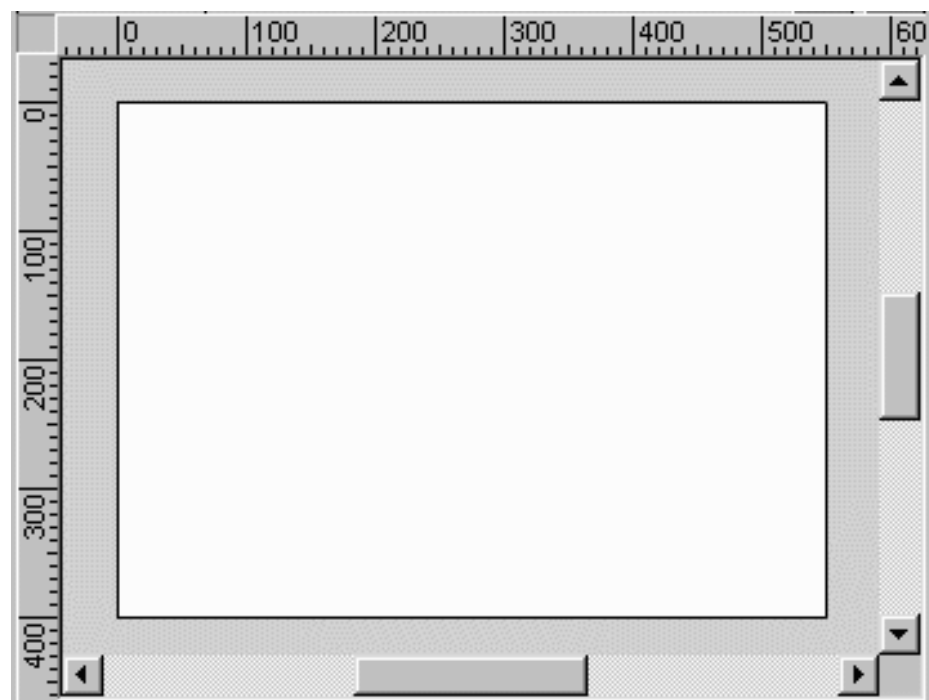
If you want to use a grid to assist in creating your animations, input the resolution of the grid (from 7.20 to 288 square pixels) and check Show Grid. Select View/Snap to force Instances to snap to the Grid.

Colors

{Click} the swatch for Grid and Background to set the Grid color and Background color.

Ruler Units

Select from Inches, Inches (decimal), Points, Centimeters, Millimeters, or Pixels to designate the units of Flash's ruler. We recommend using pixels because a pixel is the native unit of your computer monitor. To activate Flash's Ruler, select View/Rulers from the main menu. Rulers appear to the top and left side of the Stage.



Select View / Rulers to display Rulers around the Stage. Once you've set your Movie Properties, the next step is to create the Symbols you'll use to build your animation. In this case we've created two Symbols, Ball (a gradient filled circular shape), and Floor (a skewed, gradient filled rectangular shape). We'll begin with the most traditional form of animation, frame-by-frame.

Frame-by-Frame Animation

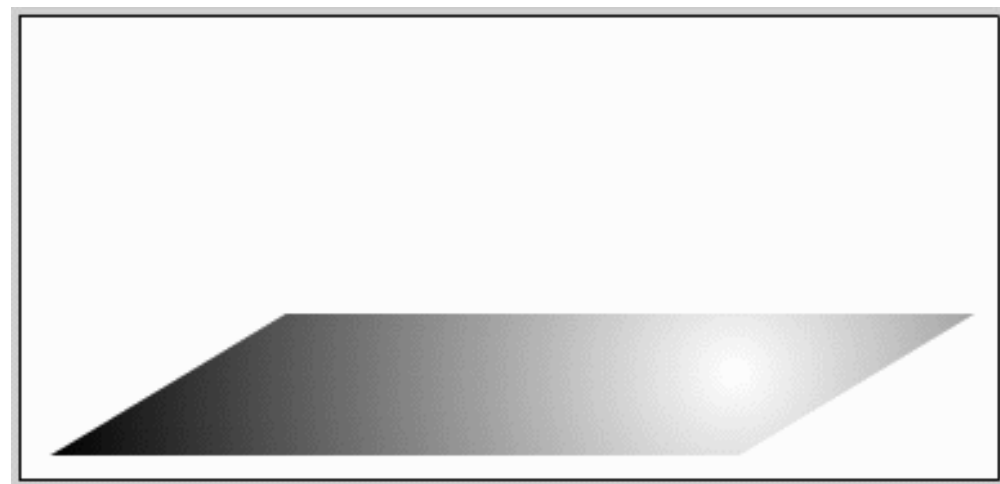
Frame-by-Frame animation involves creating each individual key frame manually. In other words, we have to create a key frame in each frame, and position and modify the Instance in each frame. To insert a new key frame, select the frame you wish to place a key frame in by {Clicking} inside the frame on the Timeline. Select Insert/Keyframe from the menu or press F6. Flash duplicates the previous key frame. In frame-by-frame animation, you create a sequence of key frames, one after the other.

Our first Layer contains an Instance of a Symbol called Floor. We need to place a floor on the Stage that the ball will bounce off of. Because the floor will not move, we only need to create one key frame and extend the information within that key frame to a

particular range of frames, in this case 10 frames. To do so, we rename the first Layer (Layer 1), floor, and place an Instance of the Floor Symbol on the Stage. The key frame is filled in frame 1.



Layer 1 renamed floor. The key frame in Frame 1 is filled because we've dragged an instance of the Floor Symbol on to the Stage.



The Floor Instance placed on the Stage.

To extend the key frame, simply *{Click}*, hold (do not release the mouse button), and drag to select frames 1 to 9. *{Release}* the mouse button and select Insert/Frame from the main menu or press [F5]. Frames 1 through 10 are grayed, indicating that the information in the last key frame to the left, in this case the key frame in frame 1 is duplicated through frames 2 to 10.

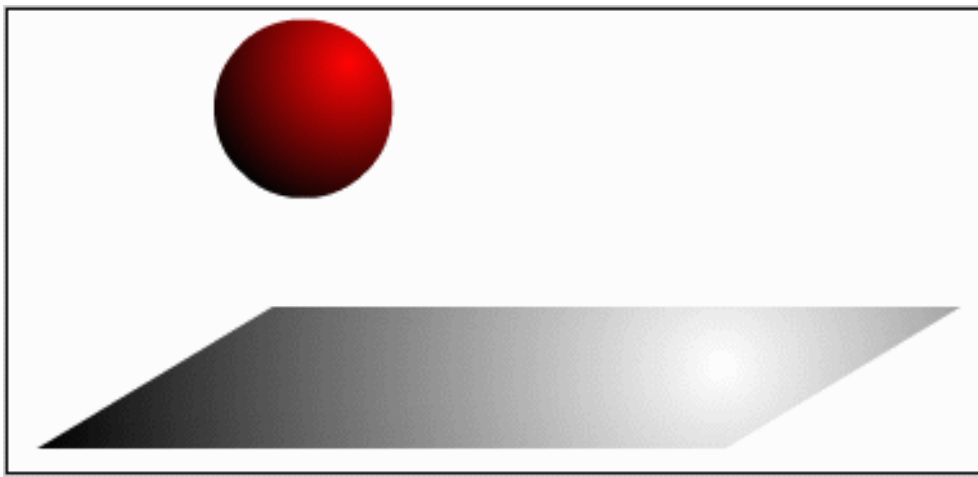


To extend the range of the key frame to frame 10, ***{Click}***, hold, and drag to select Frames 1 through 9.



Select Insert/Frame or press [F5] to extend the range of the key frame. Frames

1 through 10 are gray in color.



You can drag the Frame Marker (or Scrub) back and forth through the range of Frames. The Floor Instance is displayed in every frame.

Next we need to create a Layer for the Ball and animate it.

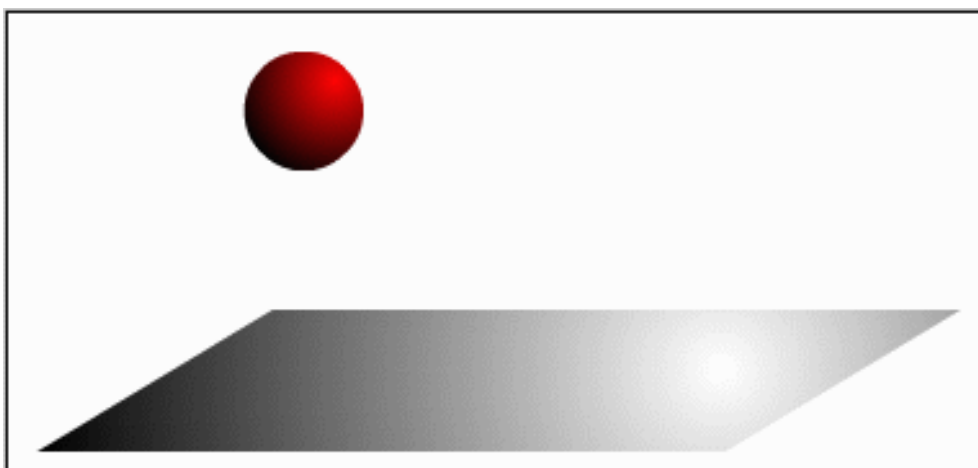
{Click} the Add Layer button . A new Layer is created in the Timeline above the currently selected Layer. Because we are going to animate the Ball on this Layer we rename the Layer to ball. You'll notice that by default, Flash extends the range of the new Layer to 10 frames. The range of Frames is not gray in color because the key frame at Frame 1 is still empty. We need to drag an Instance of the Ball Symbol onto the Stage.



The new Layer duplicates the frame range of the previous Layer.

Dragging an Instance of the Ball Symbol on to the Stage with the ball Layer selected fills the key frame at Frame 1. As we add key frames, Flash automatically duplicates the previous key frame. We have to keep this in mind as we produce our animations. In our case, the Instance is too large for our animation. Before we create new key frames we need to reduce the scale of the Instance in the initial or first key frame.

The Ball Instance is too large for our animation.



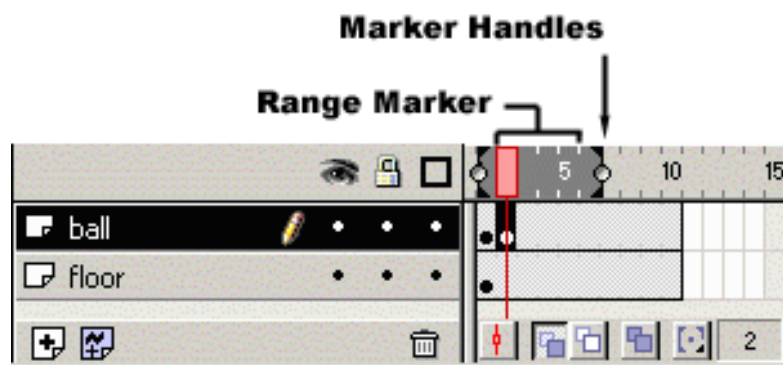
The Ball Instance reduced in scale.

Next we need to create the following, or next, key frame. We select frame 2 in the Timeline and select Insert/Keyframe or press [F6]. The new key frame copies the information from the previous key frame. You might wonder why we had to reduce the size of the Ball Instance previous to creating the next key frame. Because Flash bases the new key frame in a Layer on the previous key frame, had we not reduced the size of the Instance, we would have to do so twice, once in the first frame and again in the second frame. Taking small steps and simple planning can reduce production time in Flash.

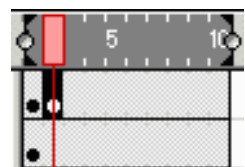
We need to move the Instance toward the Floor and slightly to the right to simulate the ball moving down and forward as it falls. When we animate frame-by-frame, it's a good idea to enable Onion Skinning.

Onion Skinning

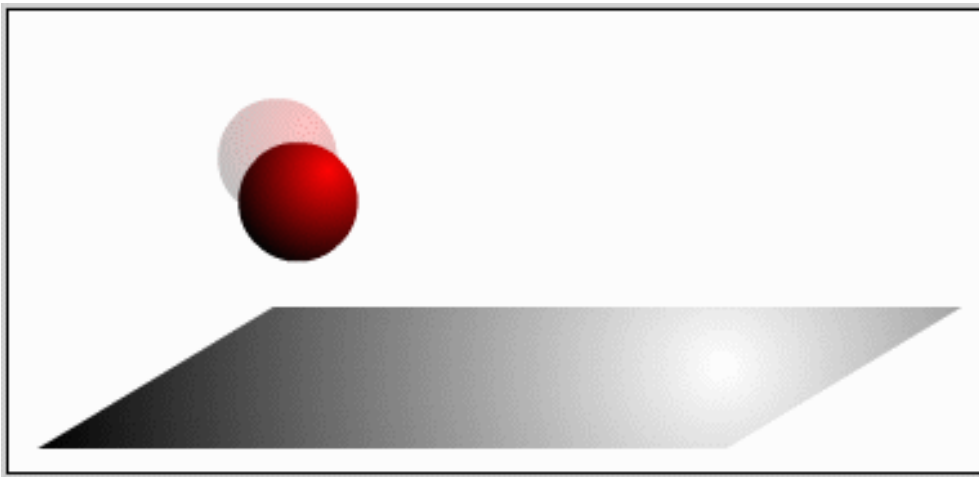
In traditional animation, animators would draw the contents of each frame on onion skin paper, a thin, translucent paper which permits them to see the state of the previous key frame or frames to use as a reference. To enable Onion Skinning in Flash, {Click} one of two Onion Skin buttons, Onion Skin , or Onion Skin Outline  at the bottom of the Timeline. Onion Skin presents us with a translucent reference of previous frames while Onion Skin Outline presents each frame as an outline. Because everything is in the same location at this time, the effects of enabling Onion Skinning are not immediately apparent. Before you position the Instance in Frame 2, you should adjust the range of frames that Flash will display as Onion Skinned. The moment you enable Onion Skinning, a Range Marker pops up at the top of the Timeline.



{Click}, hold, and drag the Marker Handles to adjust the range of frames. In this case, we want the entire range of 10 frames to be onion skinned.

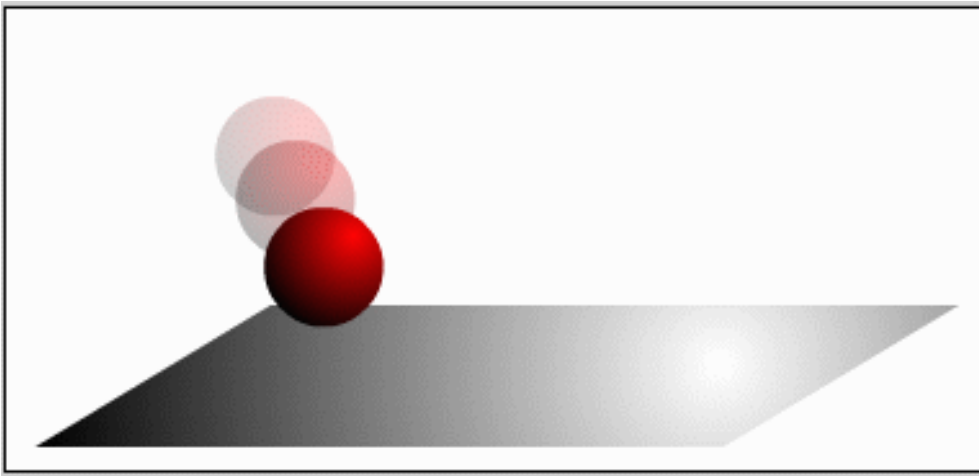


Now we can begin to position the Ball Instance in Frame 2. Drag the Instance below and slightly to the right. You'll see the position of the Instance in Frame 1 when you release the mouse button. You can use the position of the Instance in Frame 1 as a reference to properly position the Instance in Frame 2.



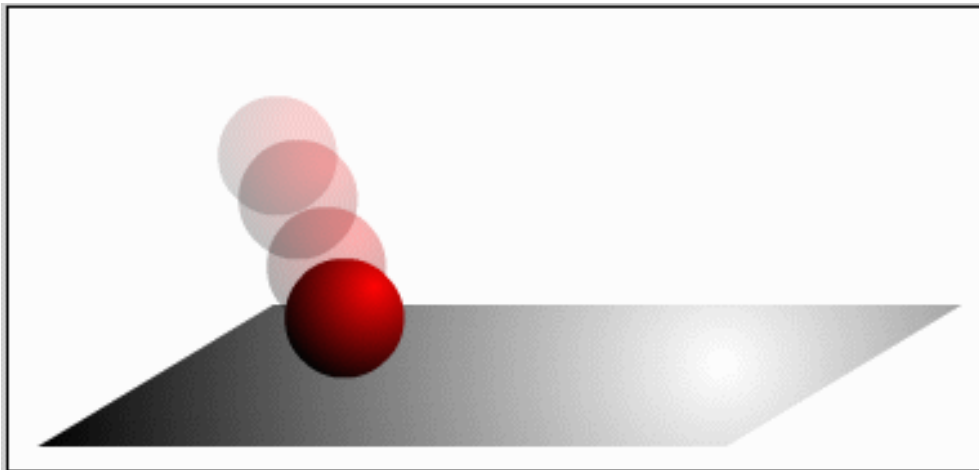
Positioning the Instance at Frame 2.

Select Frame 3 in the Timeline and insert a key frame. The previous key frame is copied. Position the Instance in Frame 3. With onion skinning enabled, you should see the position of the Instance in Frames 1 and 2.



Positioning the Instance at Frame 3.

Select Frame 4 in the Timeline and insert a key frame. The previous key frame is copied. Position the Instance in Frame 4. With onion skinning enabled, you should see the position of the Instance in Frames 1, 2, and 3. As you create key frames the end of the Range Marker may shift. Simply adjust the Range Markers appropriately.

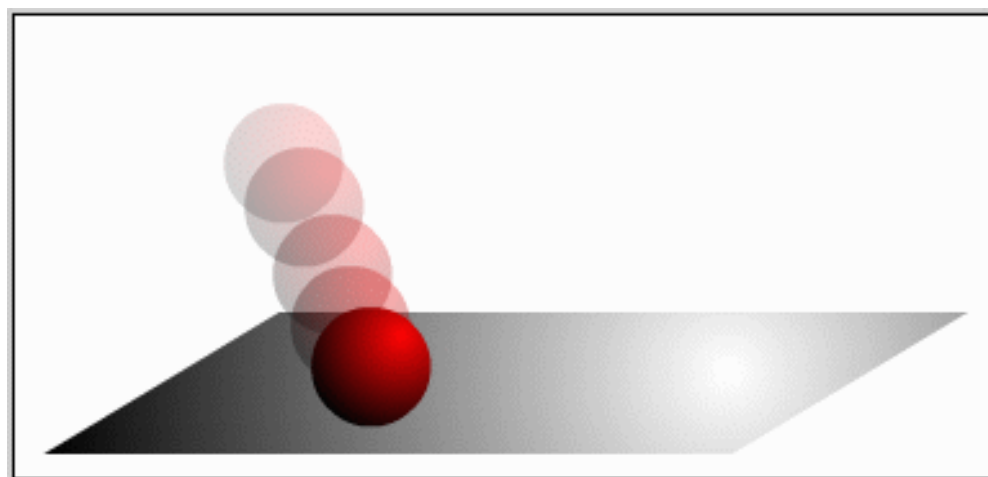


Positioning the Instance at Frame 4.

Select Frame 5 in the Timeline and insert a key frame. The previous key frame is copied. Position the Instance in Frame 4. With onion skinning enabled, you should see the position of the Instance in Frames 1, 2, 3 and 4. You may find that as you get closer to the Floor Instance, you might select it by mistake. To avoid selecting the Floor Instance, lock the Floor Layer.

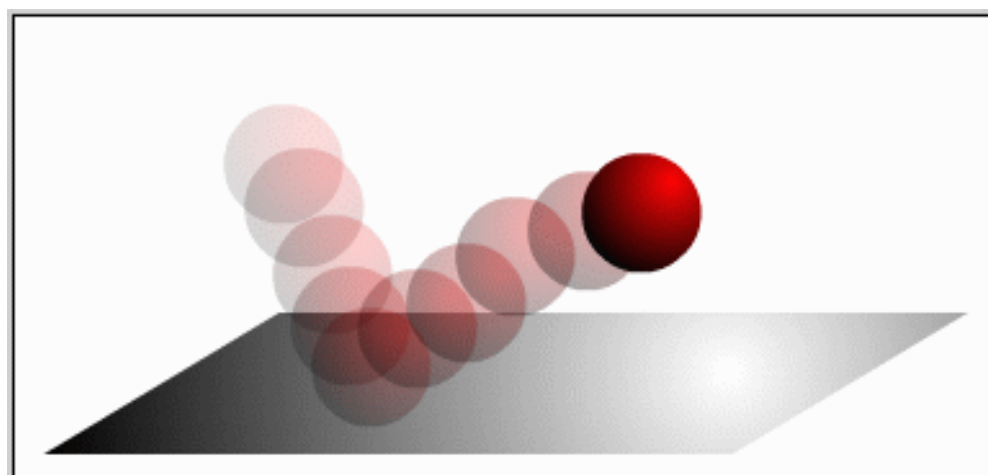


Locking the Floor Layer prevents you from accidentally selecting and moving it.



The ball hits the floor at Frame 5. Now we need to move it up.

Repeat the process of adding a key frame and positioning the Instance accordingly in Frames 6 through 10. At Frame 6, the ball should begin to go up. In Frames 7 through 10, the ball should continue to raise, and slowly arc, and pull to the right as it raises, being subject to gravity.



Each frame in our 10 frame, frame-by-frame animation.

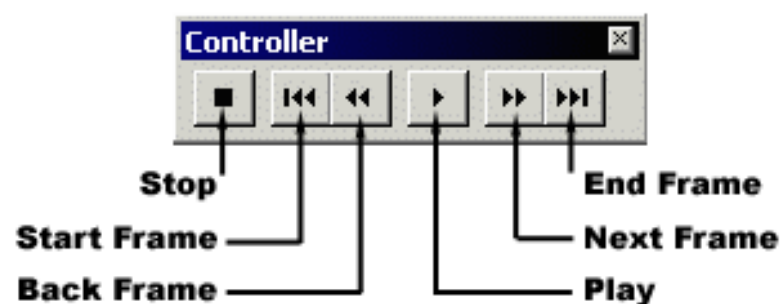
Viewing your Animation

To watch the animation, press the [ENTER] key. The animation will play from the point where the scrub is until the end of the animation. Keep pressing [ENTER] to view the animation again and again. To avoid confusion, at this point you might want to disable Onion Skinning by {Clicking} on the Onion Skin button again. You can also, {Click}, hold, and drag the scrub over the range of frames to view your animation.

Open the file `OnionSkin.fla` from the `Chapter06` folder to examine this animation.

The Controller

You can also use the Controller to view your animation. Select Window/Controller. The Controller Palette pops-up.



{Click} the Play button to play the animation. To move to the beginning of the animation, {Click} the Start Frame button. To move to the end, {Click} the End Frame button. To advance a single frame, {Click} the Next Frame button. To move back a single frame, {Click} the Back Frame button.

Tweaking the Animation

Our simple animation may need some tweaking or fine-tuning. There are a number of problems. First, in our case, the ball hits the floor to close to edge. We will need to reposition the Ball Instance in each frame. Rather than doing so on a frame-by-frame basis, we can move each frame all at once by enabling Edit Multiple Frames. Press the Edit Multiple Frames button  beside the Onion Skin controls.

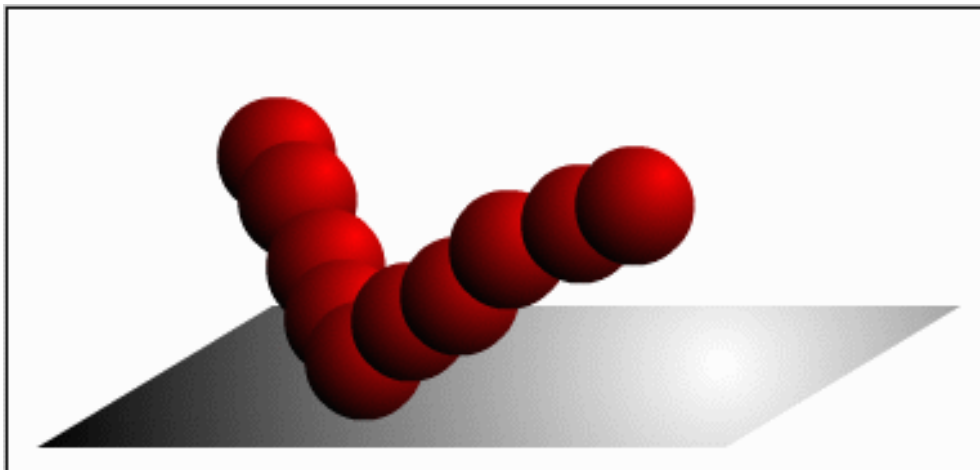
Edit Multiple Frames

{Clicking} the Edit Multiple Frames button  in the Timeline causes Range Markers to appear at the top of the Timeline, just as with the Onion Skin controls. Drag the range over the desired parts of the animation where you wish to select multiple keyframes.



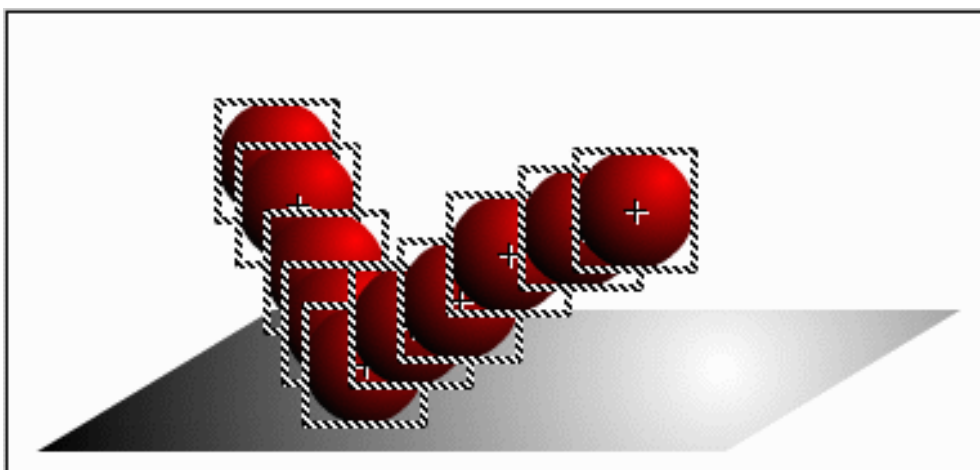
With Edit Multiple Frames enabled, the full range of 10 keyframes is designated by the Range Markers.

On the Stage, every key frame within the range is displayed.



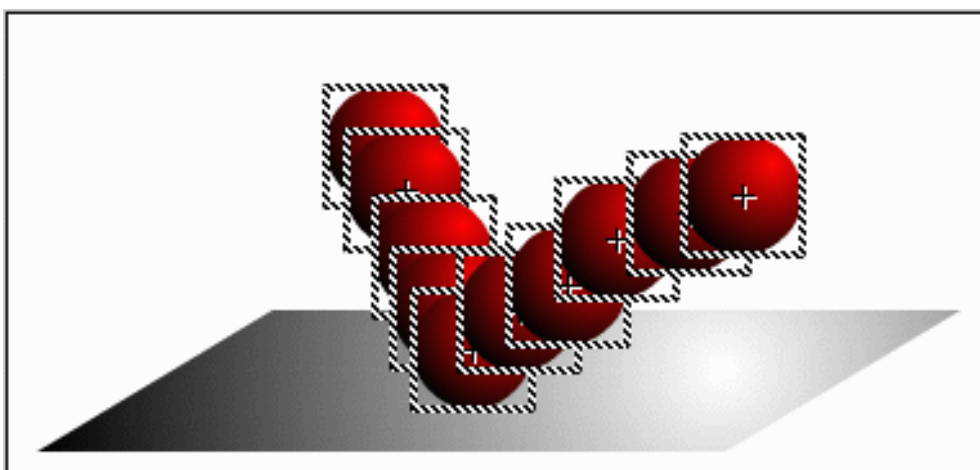
Every keyframe within the range in is displayed on the Stage with Edit Multiple Frames enabled.

To affect every key frame simultaneously with Edit Multiple Frames enabled, we have to select each key frame. To select all key frames, select Edit/Select All. Selection boxes surround the Instance in each key frame. Because the Floor Layer is locked, it remains unselected.



10 key frames are selected simultaneously.

Next we need to move the Instances to a better position. With each key frame selected, use the arrow keys on the keyboard to reposition the Instances in each key frame relative to the location of the Floor Instance. The entire block of Instances in each key frame move simultaneously, keeping the motion intact. We move the Instances closer to the middle of the Floor.



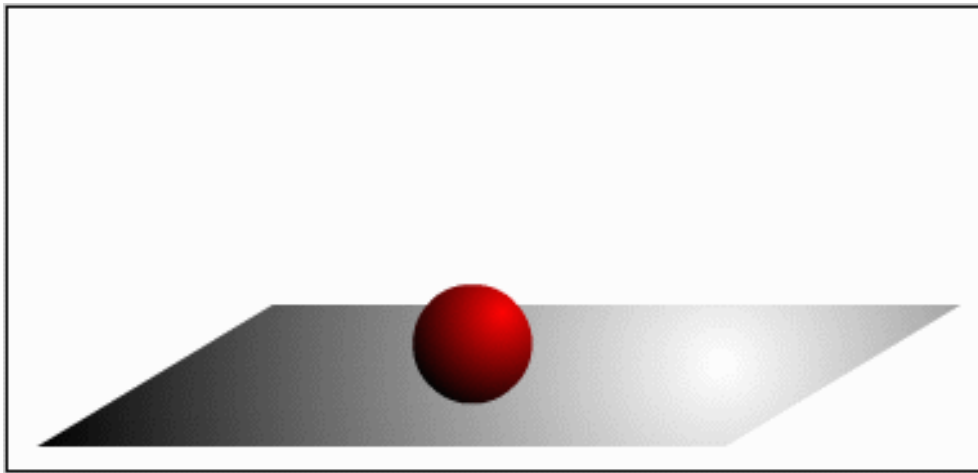
The Instances in each key frame are repositioned relative to the Floor Instance, but remain in position relative to each other.

Open the file *MultipleFrames.fla* from the *Chapter06* folder to examine this file. Enable *Edit Multiple Keyframes*.

It's vitally important to disable Edit Multiple Frames once you've completed modifying a set of multiple key frames by *{Clicking}* on the Edit Multiple Frames button again. If you forget to disable Edit Multiple Frames, what the Stage displays can get very confusing. Any Instance in a key frame, within the range, is displayed, regardless of what particular frame you are on.

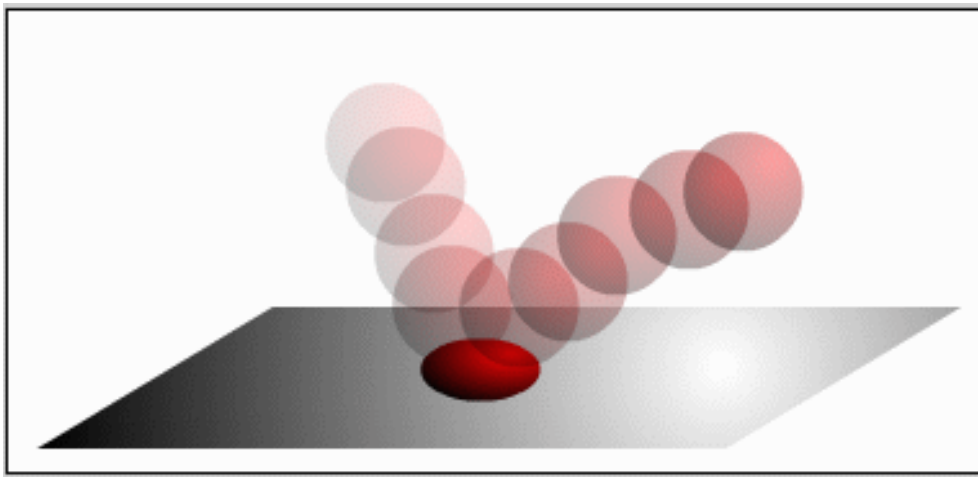
Our next problem has to do with what should happen when the ball makes contact with the floor, and what should happen to the ball on the way up, having bounced off the floor.

Frame 5 is our point of impact. Let's look at the Instance on the Stage at Frame 5.



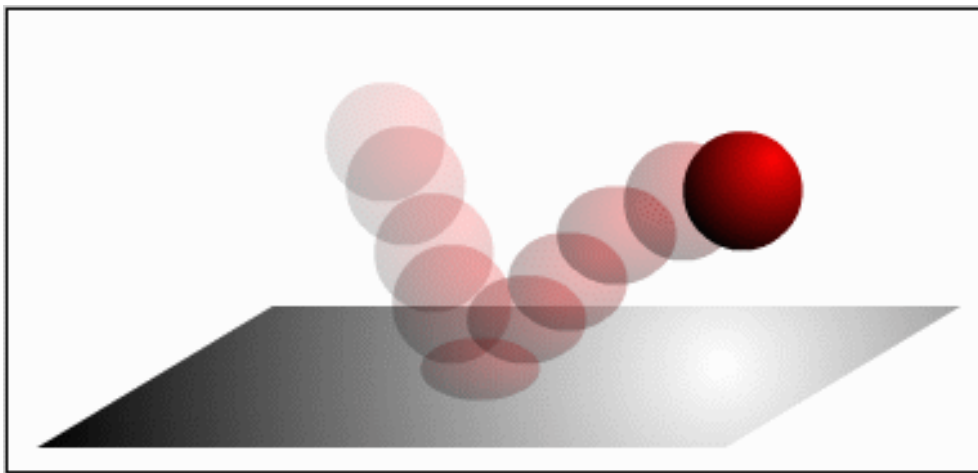
The ball appears to be floating above the Floor. If we wanted, we could build another Layer and simulate a shadow, which would reduce in size as the ball descends and increase in size as the ball ascends. However, the main problem is in the shape of the ball. Unless the ball is made of concrete or metal, the shape of the ball will deform on impact and gain its natural shape as it bounces back. We should deform the ball at Frame 5. To do so, we select Frame 5 on the Timeline. The Instance should be selected. If not, with the Frame Marker at Frame 5, we *{Click}* on the Ball Instance to select it.

We need to squash the ball by using the Scale Modifier . We don't want to reduce the size uniformly, rather, we need to grab the anchor at the top-center of the Scale box and deform the ball so it appears to get wider. As we do so, we might want to slightly reposition the ball and enable Onion Skinning for accuracy.



At the point of impact, the ball's shape deforms.

In order to make the animation just a little more realistic, we'll need to deform the ball in Frames 6 through 8, making sure that the ball eventually snaps back to its natural shape.



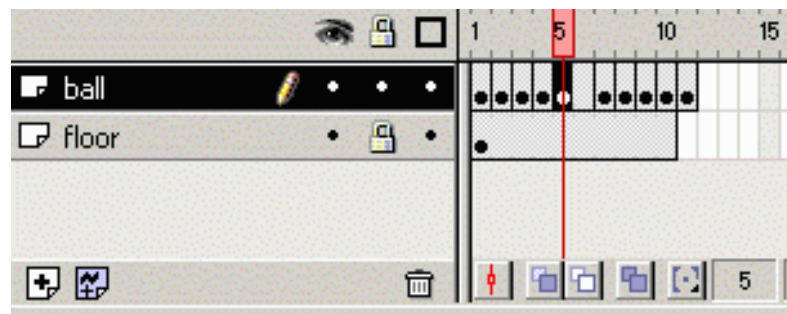
The ball deforms on impact and regains its shape as it bounces off the ground.

When you play the animation now, it should look a little more realistic. However, one major problem remains when we think about timing. We are using only one frame to squash the ball when we should try to use more. As a ball hits the ground, it squashes because it's hitting a solid surface and motion has ceased. The ball will spend a little time squashing as it hits the ground until it gets to a point where its own mass resists the squashing effect and pushes back, which results in the ball bouncing off the floor. We need to add additional keyframes to slightly prolong the point of impact.

Open the file *Squash.fla* from the *Chapter06* folder to examine this file.

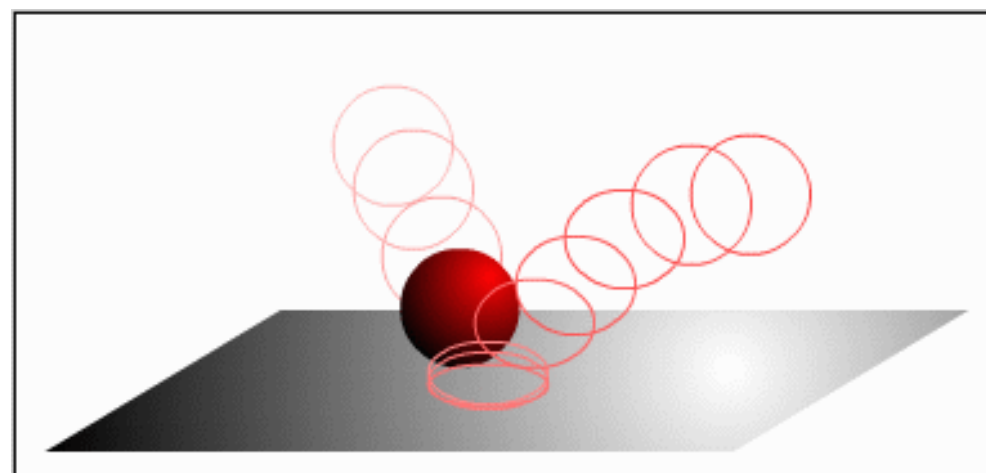
Adding Frames

To add additional frames, simply select the frame in the Timeline where you want to place additional frames and select Insert/Frame or press [F5]. A frame is added.



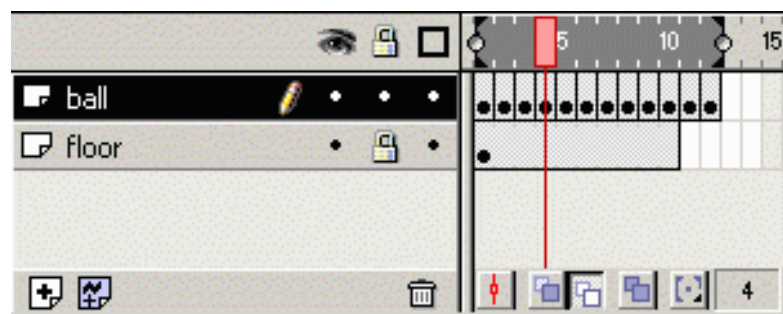
Selecting Frame 5 and pressing [F5] or selecting Insert/Frame, inserts an additional frame after the currently selected frame in the Timeline.

We need to add two additional frames. Simply repeat the process. When we have added two additional frames, we need to insert key frames into each new frame and modify the Instance. In this case, at Frames 5 through 7, the Instance will remain at the same location, but will continue to squash or compress in shape.

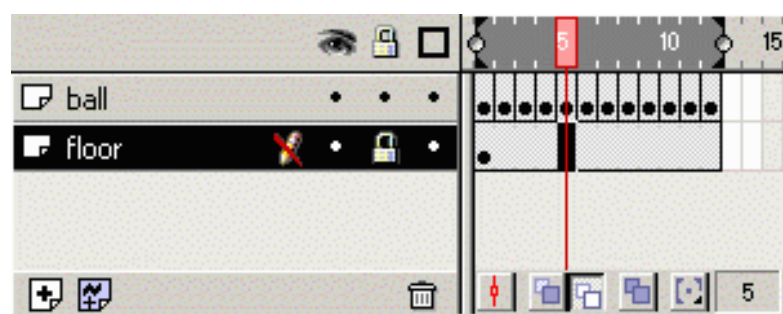


By using the Onion Skin Outlines control, we can see that the squashing process of the Instance takes up 3 frames.

Because we only added frames to the Ball Layer, we'll need to do the same to the Floor Layer to ensure that the floor remains visible throughout the animation.



Adding frames on one Layer does not add frames on other, or all Layers.



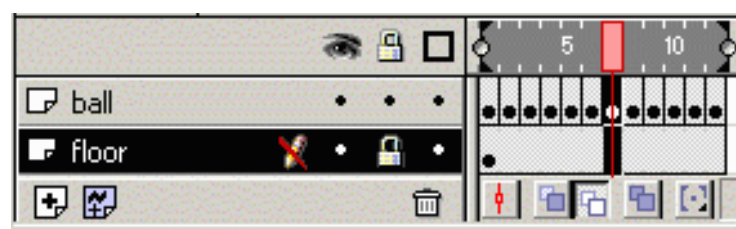
We need to select a frame in the other Layer and add frames accordingly.

When we play our animation now, we notice that before the squashing sequence was too quick, now, because of the small amount of frames we are using, the squashing sequence is too long. We'll need to remove a frame.

Removing Frames

The process of removing frames from the Timeline is similar to that of adding frames. Select the frame you want to remove and select Insert/Delete Frame or press [SHIFT] + [F5]. The frame you selected is deleted. If you continually select Insert/Delete Frame or press [SHIFT] + [F5], frames are continually deleted.

Rather than removing a set of frames from each Layer individually, you can select a frame, or group of frames, across Layers and add or remove them.

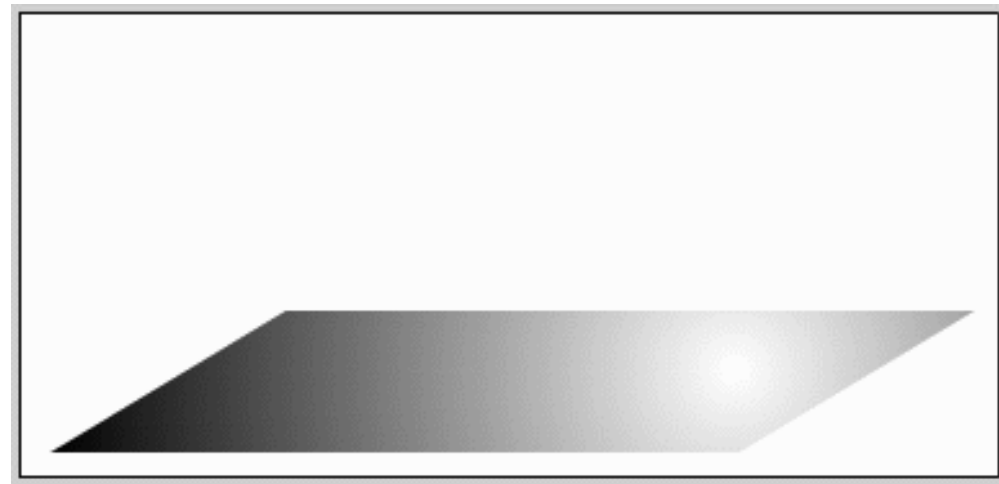


Frame 7 is selected across both the Ball Layer and the Floor Layer.

With the frame removed from both Layers, the animation is improved.

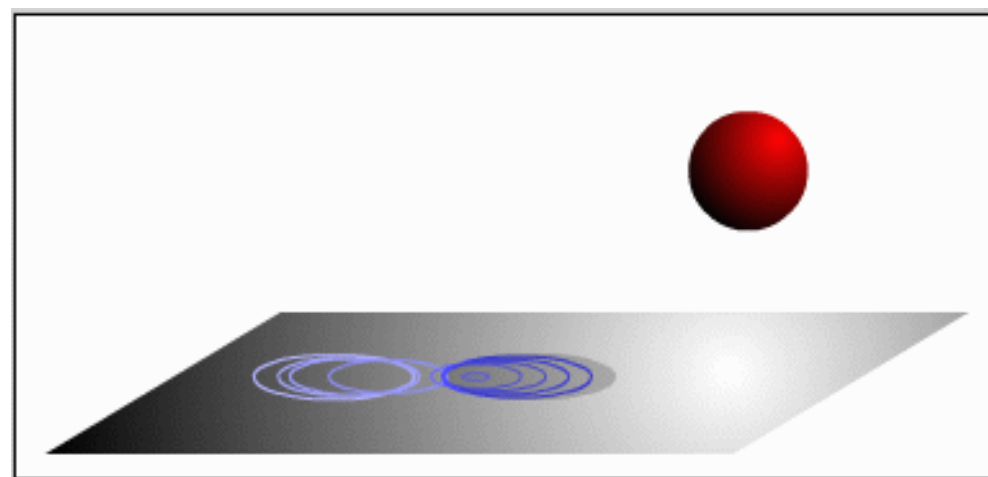
Adding a Shadow

Our final step is to add a shadow. To do so, we have to think about the light source. If we examine the floor, the gradient pattern dictates that the light source is above and to the right of the scene. To create a realistic shadow animation, we have to consider where the light is coming from.



The Floor. Because of the gradient fill, the light source is clearly above and to the top right of the scene.

As the ball descends, the shadow will decrease in size. Through Frames 1 to 5, we need to place the shadow in relation to the main light source and scale it in each frame. The shadow has to follow the ball and the opacity or alpha value of the shadow increases as the ball gets closer to the floor. Through frames 6 to 10, the shadow will increase in size as the ball rises and the opacity of the shadow will decrease the higher the ball rises.



The path of the shadow as seen with Onion Skin Outlines enabled.

Having squashed the ball and added a shadow, the animation is more realistic than our initial build. You'll find that building animations in Flash involves a lot of tweaking to achieve the affect you are after. With continued use, building and tweaking animations will become second nature.

Open the file Shadow.fla from the Chapter06 folder and view the animation.

Exercises

1. Create a bouncing ball animation as described in this chapter.
2. First create the Symbols, a ball and a rectangle (for the floor). Call them ball and floor.
3. Next, create the three (3) Layers. Name the bottom Layer floor. Name the second Layer ball. Name the third Layer shadow.
4. Select the floor Layer. Drag an Instance of the floor Symbol onto the Stage.
5. Use the Rotate and Scale Tools to skew and squash the rectangular floor.
6. At this point, the floor Layer has only one key frame and range of one frame.
7. Extend the range of frames to ten (10) frames in total.
8. Lock the floor Layer.
9. Select the ball Layer.
10. Drag an Instance of the ball Symbol onto the Stage.
11. Place the ball Instance above the Stage and to the left side.
12. Extend the range of frames of the ball Layer to ten (10).
13. Enable Onion Skinning. Extend the Onion Skin markers to the full range of frames.
14. Add key frames, one after the other, in the ball Layer.
15. Position the Instance in each key frame accordingly, so that the ball moves toward the floor, hits the floor at Frame 5, and bounces off the floor.
16. Play your animation.
17. Tweak the animation by squashing the Instance in Frame 5 and squashing less so in frames six (6) through eight (8) so that the ball regains its shape.
18. Lock the ball Layer.
19. Move the shadow Layer below the ball Layer (but above the floor Layer).
20. Drag an Instance of the ball Symbol onto the Stage.
21. Use the Special settings in Instance Properties to change the color of the Instance to black.
22. Use the Scale and Rotate Tools to flatten the ball so that it sits properly along the floor as a shadow.
23. Animate the shadow on a frame-by-frame basis.

Review Questions

1. Describe Persistence of Vision.
2. What are key frames?
3. What type of information about an Instance is stored in a key frame?
4. What are in-between frames?
5. How do you set the frame rate and dimensions of your Flash movie?
6. How do you edit multiple key frames
7. How do you view multiple frames?
8. To avoid moving an Instance on another Layer as you build your animation, what can you do to the Layer?
9. How do you add and remove frames in the Timeline?

Summary

As a result of this chapter, you should be able to

- Understand Animation basics
- Use the Timeline
- Create Frame-by-Frame Animations



In addition to traditional frame-by-frame animation, Flash supports tweening or in-betweening, where you setup keyframes and Flash automatically creates all the in-between frames.

OBJECTIVES

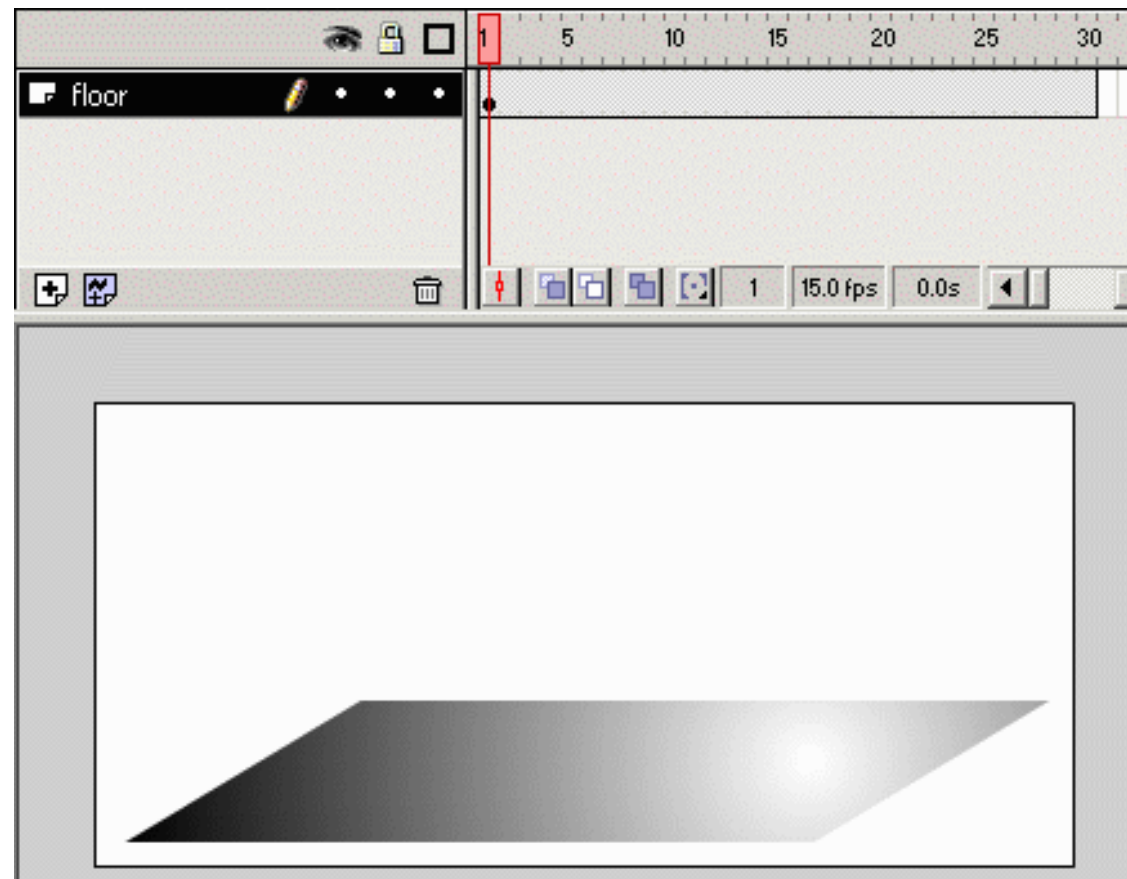
Upon completing this section, you should be able to:

1. Use Motion Tweening
2. Use Guide Layers in Animation
3. Use Shape Tweening
4. Use Mask Layers in Animation

Motion Tweening

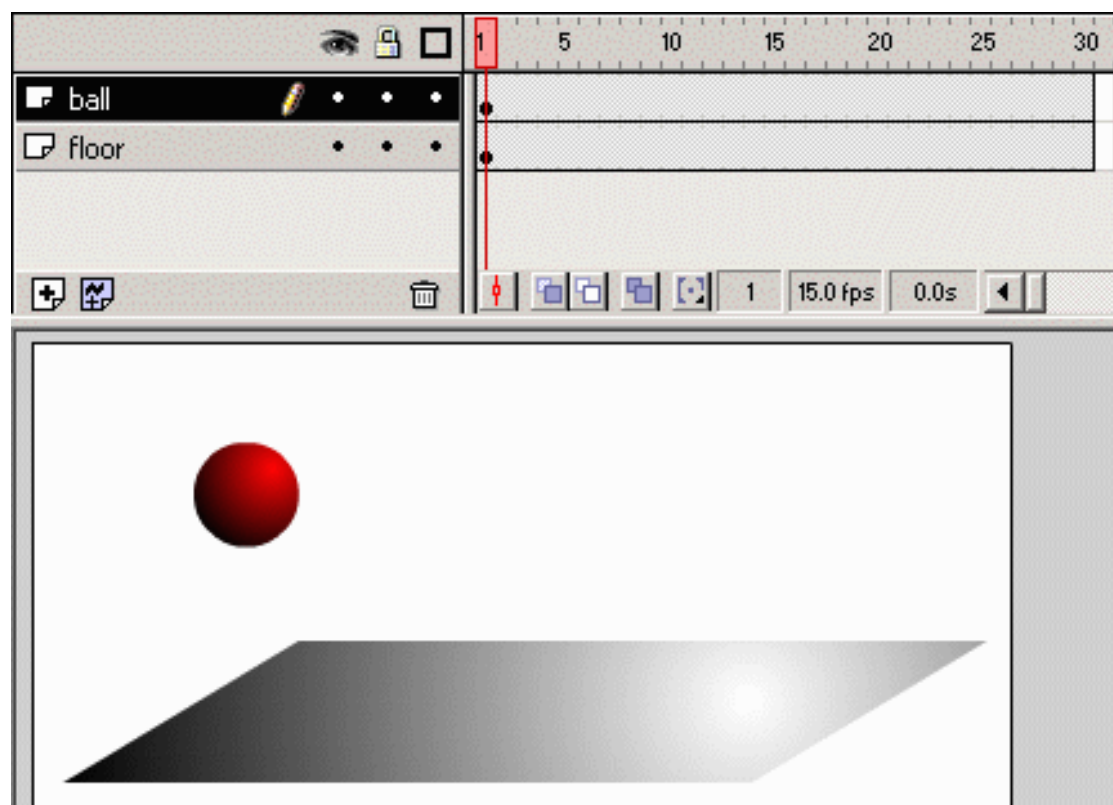
Motion tweening is the primary method of animation in Flash. The process of motion tweening involves creating at least two key frames, such as a key frame at Frame 1 and a key frame at Frame 40, and having Flash calculate and create the frames in-between. You can have multiple key frames on the same Layer, but you are limited to one Instance per Layer when motion tweening. You cannot, for instance, place two Instances in a key frame, and motion tween both. Nor can you place one Instance of a Symbol in a key frame and place another Instance of the same Symbol in another key frame on the same Layer.

We can reproduce the bouncing ball animation using motion tweening. We create a Layer for the floor, place our floor at Frame 1 in a key frame, and extend that key frame for a given number of frames, in this case to Frame 30.



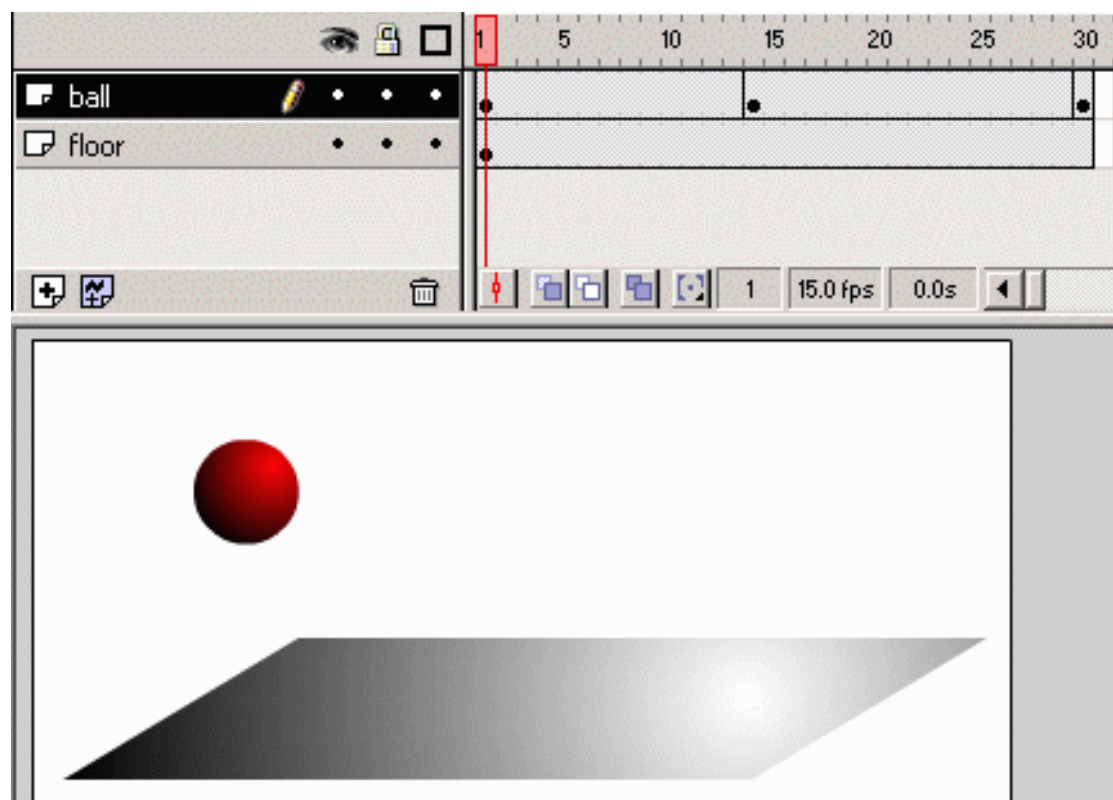
The Floor is extended to a range of 30 frames.

Next, we create a new Layer for the ball, and place an Instance of the ball Symbol on the Stage.



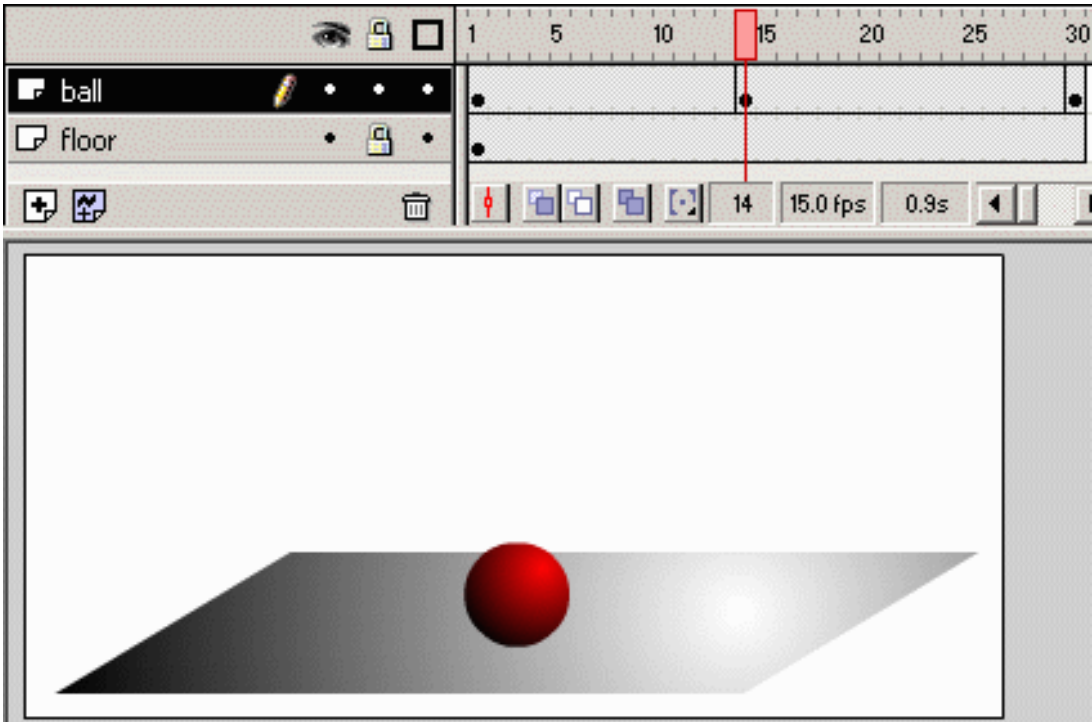
The ball is placed on the Stage.

We need three key frames for our animation. One at the beginning (Frame 1), one at the point of impact (Frame 14), and one at the end (Frame 30). As we create each key frame, the information from the previous key frame is duplicated. In other words, the Instance that we placed in the initial key frame is duplicated in subsequent key frames within the Layer. To add key frames, we select the Frame on the Timeline we want to create a key frame at and select Insert/Keyframe or press [F6].

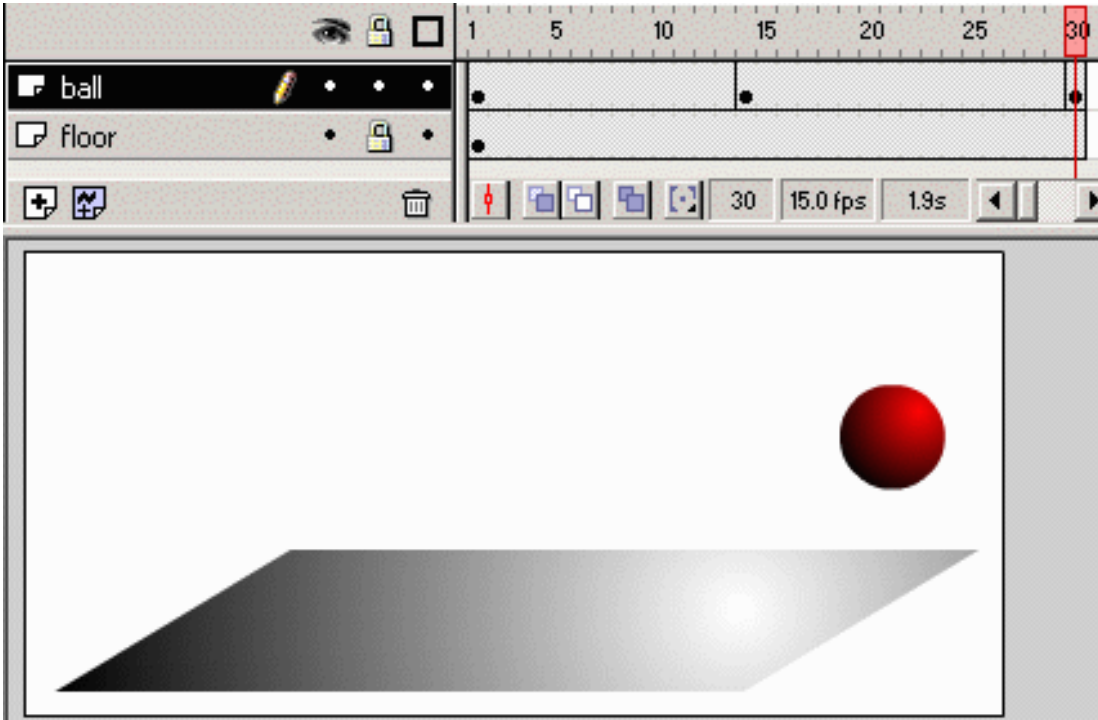


Three keyframes are placed on within the ball Layer.

Each key frame (at Frames 1, 14, and 30) is, at this point, identical. To modify the Instance in each key frame simply select the key frame you want to modify and manipulate the Instance accordingly. At Frame 14, we need to place the ball at the point of impact, and at Frame 30, we need to place the ball above the floor.



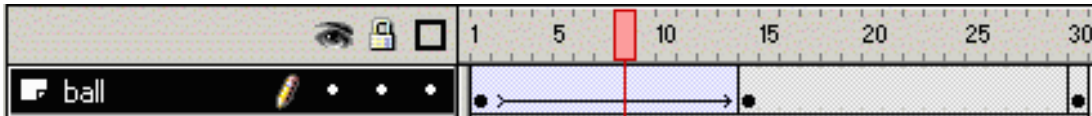
The ball placed at the point of impact. We'll squash the ball later.



The ball placed at the end point in Frame 30.

Now we can tween the animation to make Flash create all the frames in-between our key frames. Its important to understand how the tweening process works. We have only one Instance within this Layer. We did not drag another Instance of the Symbol onto the Stage within each key frame, rather, you should think of it as the same Instance at different points in time. At these different points in time, the Instance can change position, change scale, change transparency (alpha), and so forth. When we motion tween, Flash compares the key frames (or moments in time) and creates the in-between frames automatically.

To motion tween, select a point in the Timeline anywhere in-between two key frames and select Insert/Create Motion Tween or {Double-Click} the mouse. If you select Insert/Create Motion Tween, a motion tween is automatically created, without giving you control over the tween (you can modify it after). In the Timeline, the area between the key frames is colored purple and an arrow points from one key frame to the other.

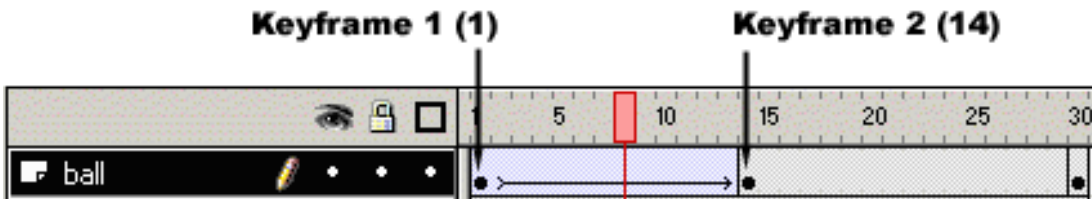


Keyframes at Frame 1 and Frame 14 are motion tweened.

If you {Double-Click} the area in between the key frames, the Frame Properties dialog-box pops up.

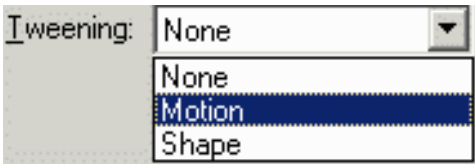


Select the Tweening Tab. By default, when you {Click} anywhere within the Timeline, the first keyframe previous to the point at which you {Click} at is the frame that modifications are applied to.

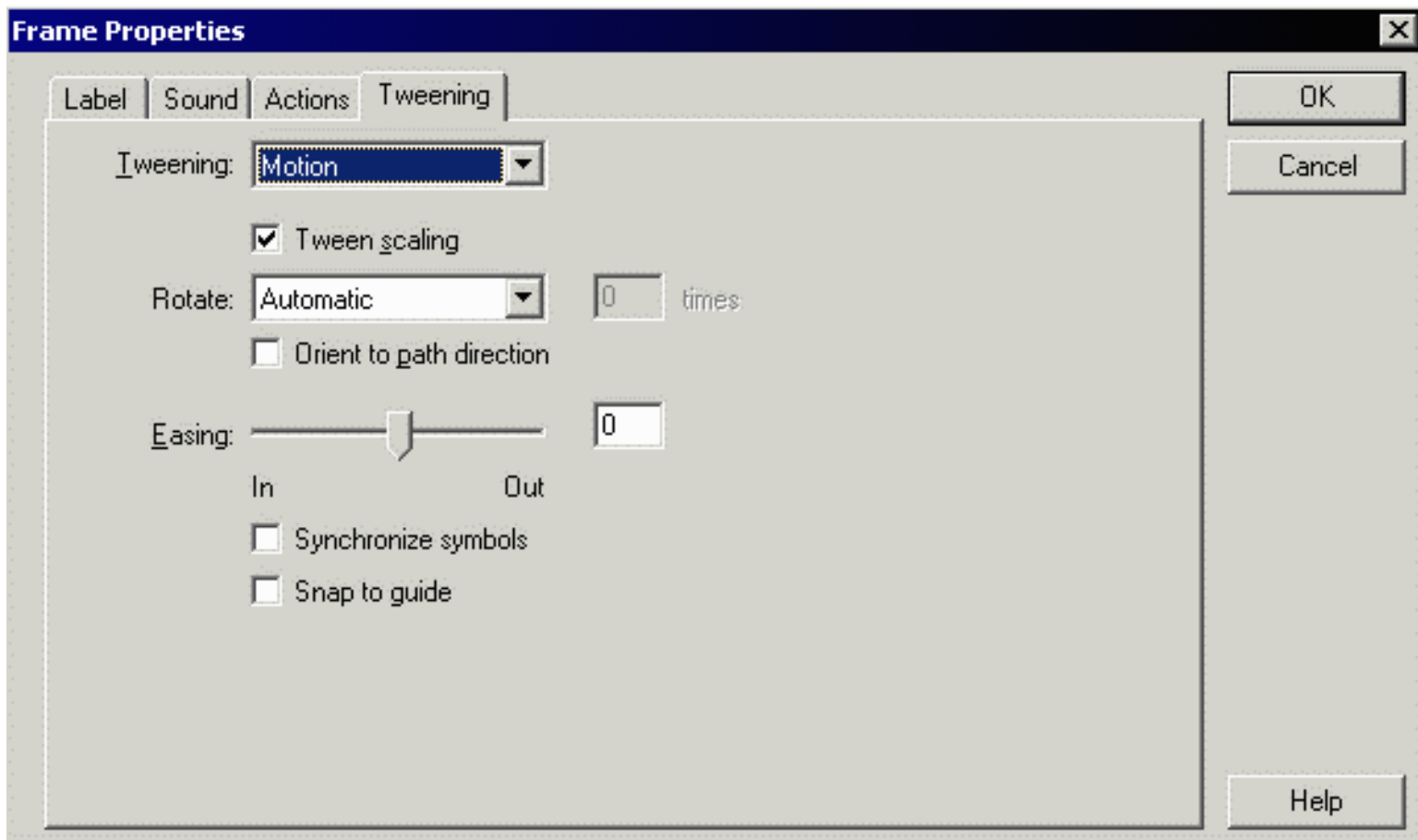


The Motion Tweening settings are applied to the key frame at Frame 1. Flash compares the state of the Instance at the key frame in Frame 1 with the state of the Instance in key frame at Frame 14 and creates the in-between frames accordingly.

From the Tweening drop-down menu, select Motion Tween.



A number of options to control the motion tween are presented in the Tweening options.

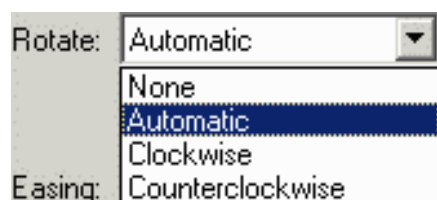


Tween Scaling ☒ Tween scaling

Tween Scaling is checked by default. This ensures that any changes in scale of an Instance from one key frame to another are tweened.

Rotate

Select from None, Automatic, Clockwise, or Counterclockwise. If you select None, any rotation you've applied to an Instance will not be tweened. By default, Automatic is selected. Automatic will tween any rotation you've applied to an Instance. Select Clockwise or Counterclockwise to have Flash automatically rotate the Instance 360 degrees in either direction. Input the amount of times you want the Instance to rotate in the times field.



Orient to Path Direction

☐ Orient to path direction

When we create motion guides for our animation, *{Clicking}* this option makes the Instance follow the path according to its orientation. We'll cover this feature later.

Easing

Easing controls the pace of the Instance from one key frame to another and is an excellent method of adding realistic motion to your animation. Easing in causes the Instance to move slowly at the start and come up to full speed. Easing out causes the Instance to slow down as it reaches the end. Drag the slider to control the amount of easing in or out that is applied to the motion tween. By default, easing is set to 0, meaning motion will be continuous, from one point to another.



Synchronize Symbols ☐ Synchronize symbols

We use this feature when we need to synchronize the timing of multiple timelines. We cover this feature in the Movie Clips chapter.

Snap to Guide ☐ Snap to guide

When we create motion guides, we can force the Instance to snap to the guide. We cover this feature later in the chapter.

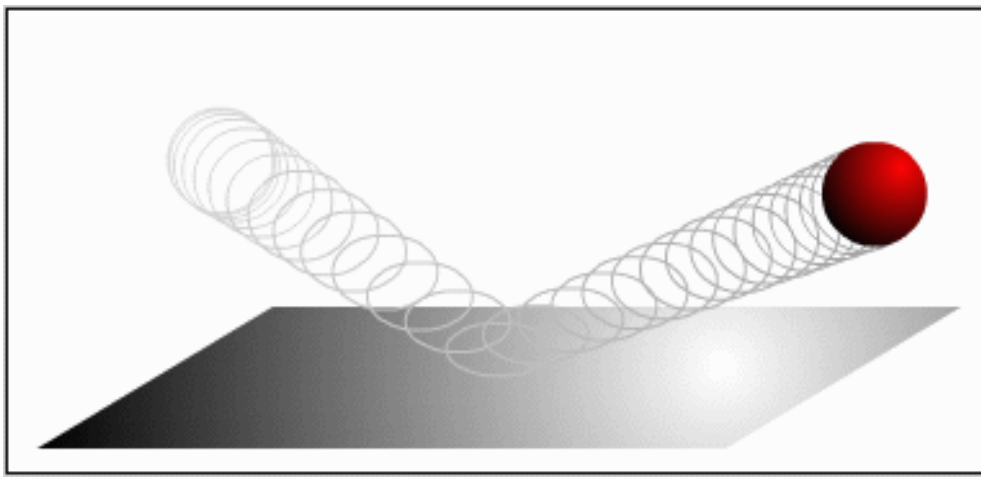
If you have already applied a motion tween, simply *{Double-Click}* the tween in the Timeline to edit tween properties.

In this case, if we think about the trajectory of a ball, it will be slowing down as it reaches the top of the trajectory, and it will speed up as it drops down to the ground. We should select Easing In for our first motion tween so that the ball speeds up as it falls to the ground. If you do not use the Easing controls (leaving Easing at the default of 0), the motion of the ball appears unnatural.

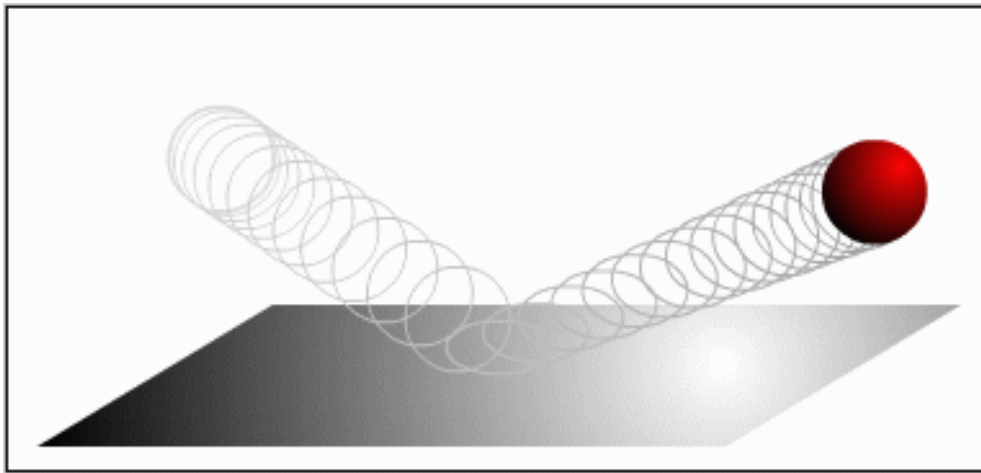
For our second motion tween, we want to ease out. Why? As the ball reaches the top of its trajectory, it will begin to slow down. Easing values often have to be quite high (from 60 to 100) to be noticeable. You'll have to experiment with the level of Easing until you create the desired effect. In addition to using the slider, you can also input values into the Easing field. Out values are positive, In values are negative. We add the second motion tween by *{Double-Clicking}* in-between Frames 14 and 30, making sure to set easing to out.

Our next step is to modify the Instance at Frame 14 by squashing it and positioning it appropriately at the point of impact. Because of the way motion tweening works, by comparing the state of the Instance in one key frame to another, the ball actually begins to squash before the point of impact. On the bounce up, the ball will regain its form, which works well. But on the way down, we don't want the ball to squash until it hits the floor. Because we squashed the Instance by modifying the scale, we merely have to modify the motion tween and uncheck the Tween scaling control

☒ Tween scaling. Flash will ignore the change in scale between the key frames, but will not ignore the change in position. Modify the motion tween by *{Double-Clicking}* in between the key frames on the Timeline or *{Double-Clicking}* the first keyframe in the motion tween sequence.

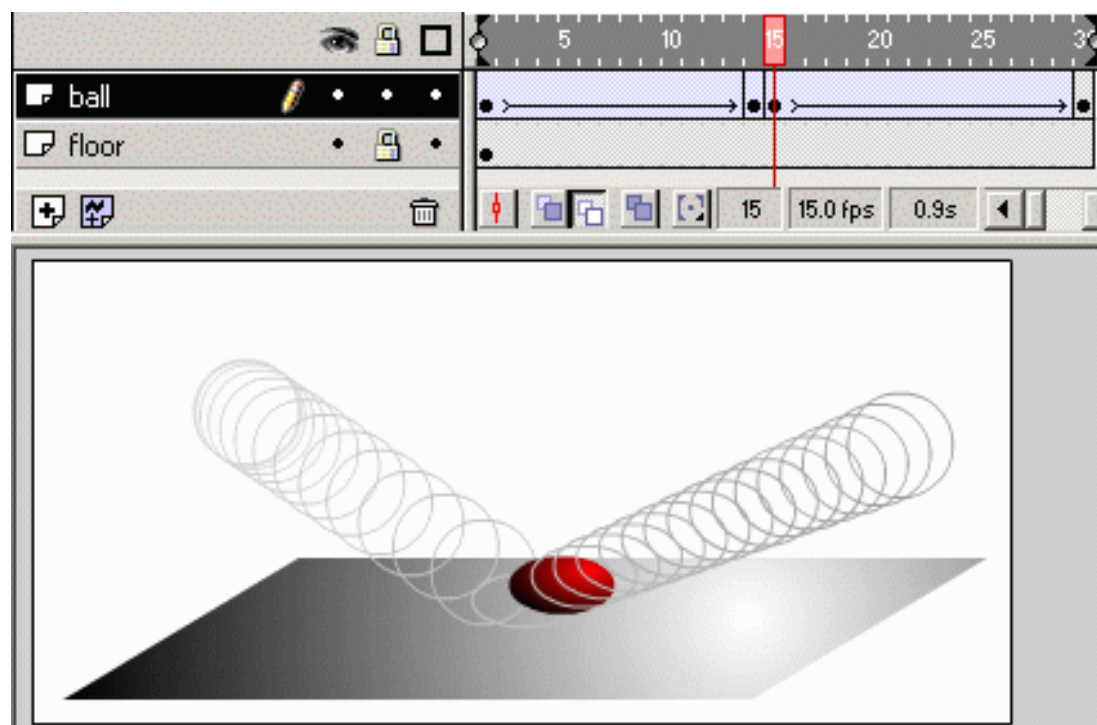


With Tween scaling enabled, the ball begins to squash before Impact.



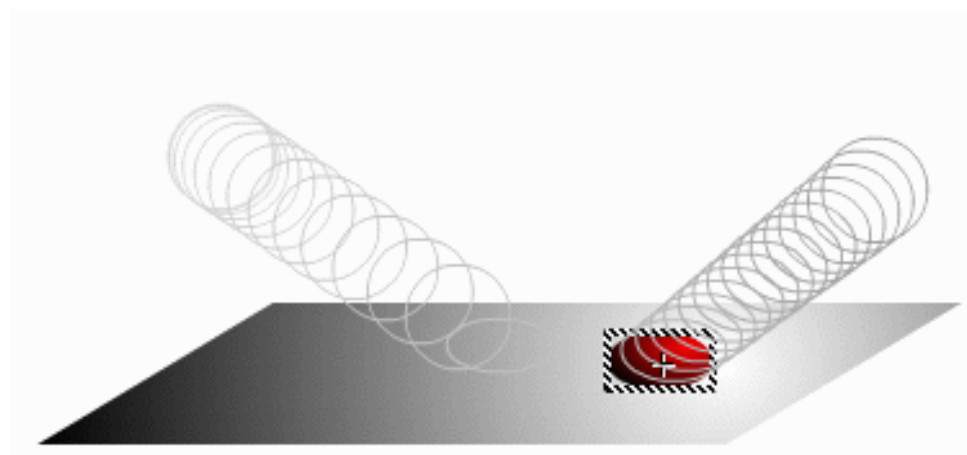
With Tween scaling disabled, the ball does not squash before impact.

Our next step is to add an additional key frame at Frame 15 to make the squash process last a little longer. You'll find that many animations require the use of a combination of approaches, in this case motion tweening and frame-by-frame. When you already have motion tween occurring in the Timeline, and you add a key frame, Flash creates the key frame containing the state of the Instance as it calculated it. In this case, it does not duplicate the previous key frame. When we create a key frame at Frame 15, it does not replicate the key frame at Frame 14, but converts the state of the Instance as it calculated it to a new key frame. As a result, we have to manually select our new key frame and position and scale the Instance accordingly. Its usually a good idea to enable onion skinning to see the affects of your adjustments.

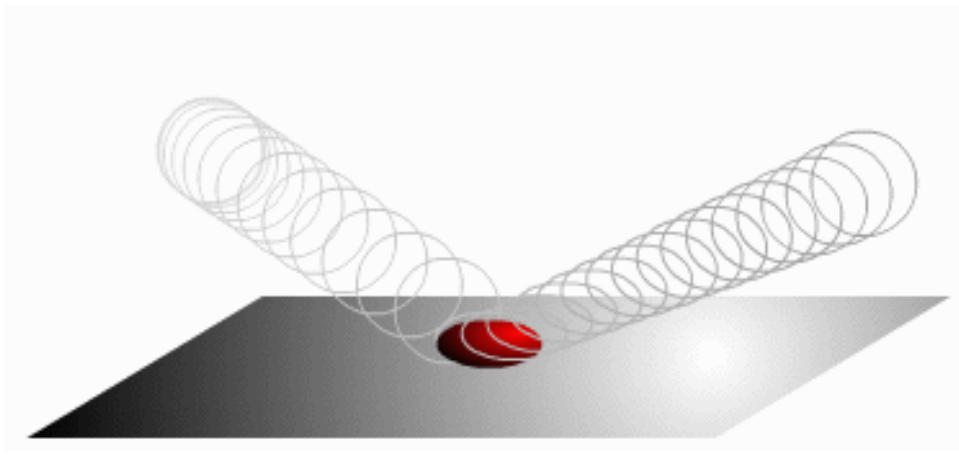


Originally, we had motion tweened the key frame at Frame 14 with the key frame at Frame 30. When we created a new key frame at Frame 15, Flash created the key frame based on the state of the Instance as it had calculated it when tweening between the original key frames.

When you adjust the Instance in the key frame at Frame 15, and you have onion skinning enabled, you can view the affects of your modifications in real-time.



With onion skinning enabled, modifying the Instance in the currently selected key frame gives a real-time preview of the affects our modifications will have.



The Instance in the new key frame positioned and scaled properly.

Because we added a new keyframe, we'll need to adjust the Easing controls to ensure that the Easing affect we had originally applied to the key frame at Frame 14 is applied to our new key frame at Frame 15. In addition, because the key frame at Frame 14 no longer needs to be motion tweened, we can *{Double-Click}* the key frame on the Timeline and change Tweening from Motion to None.

Open the file `BouncingBall.fla` from the Chapter07 folder and view the animation.

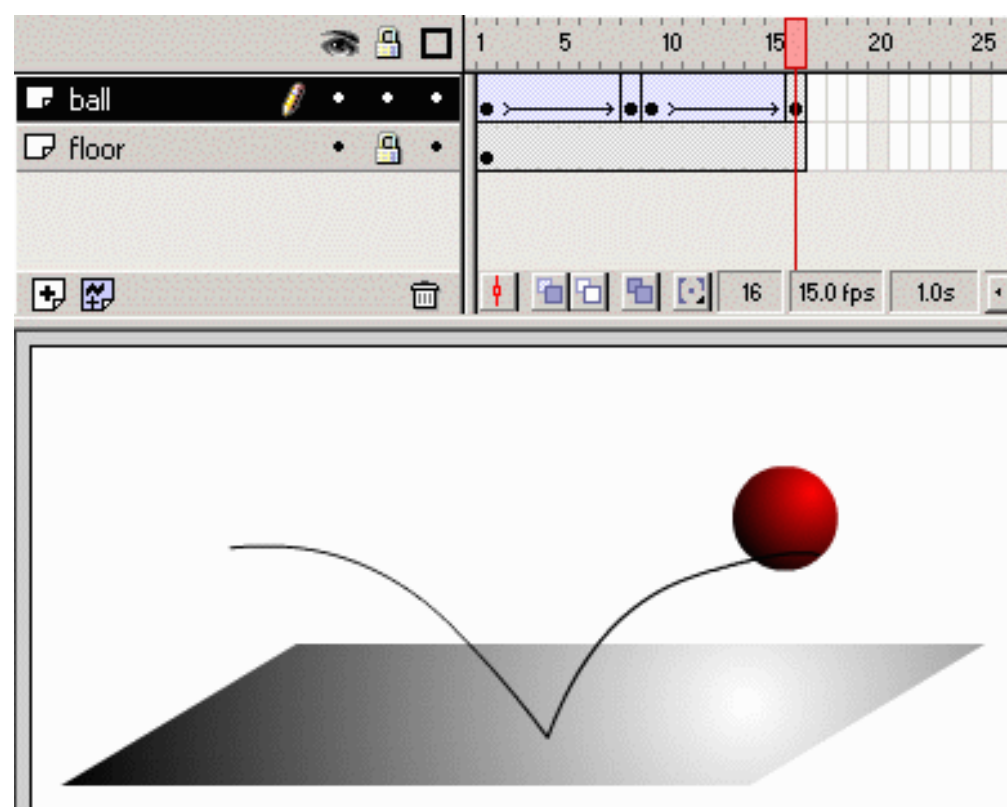
Our final step is to adjust timing. At present, the animation is much too slow. One possible solution is to increase the frame rate of the movie, but because of processor speed concerns, increasing the frame rate is not a good solution (unless you've set it quite low). If the animation is too slow, and your frame rate is set at an acceptable pace (15-20 fps), you probably have too many frames in your animation. Consider how far the ball must move on its downward trajectory and its upward trajectory, and then consider how many frames we are using to animate each trajectory. We have too many. To reduce the number of frames, you can add or remove frames using Insert/Frame or Insert/Delete Frame. Make sure you add or remove frames in between the key frames or you'll remove the key frames by mistake. You can also drag the key frames on the Timeline closer together until you are satisfied with the speed of the animation. To drag key frames, ***{Click}*** the key frame and release the mouse button. A small box should appear beside the cursor. ***{Click}***, hold, and drag the key frame again to move it to another frame. You might need to adjust easing controls as well. In addition, you can adjust the position of the Instance in each key frame until you're satisfied with the trajectory.

Open the file `BallTweaked.fla` from the Chapter07 folder and view the animation.

Motion Guides

When Flash motion tweens an Instance where a change in position has occurred, it does so in a straight path. In our bouncing ball animation for example, the ball falls to the ground in a straight line and bounces off the ground in a straight line. Using the frame-by-frame method of animation, we can make the ball fall and bounce along an arched path, but using the frame-by-frame method is a tedious solution. Fortunately, in Flash we can use motion guides (often referred to as motion paths). After we have created an animation using Motion Tweening, we can apply a Motion Guide to that animation, forcing the Instance to follow the path or shape of the Guide. Motion guides are lines that must be on the Shape level.

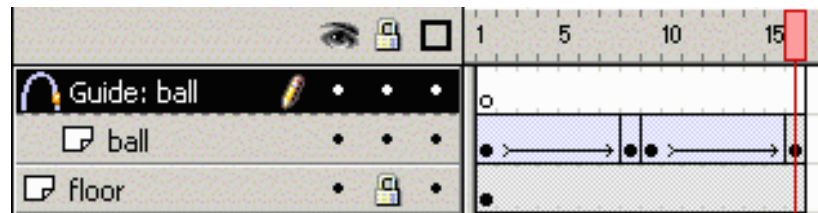
To create a motion guide, simply draw a line in the shape you wish, using the Pencil Tool. For our ball animation we need to draw a line that arcs down toward the floor and up from the floor. Once you've created a motion guide, it's a good idea to make it into a Symbol for later use. Once you've created the guide and converted it to a Symbol, delete it from the Stage.



A motion guide on the Stage.

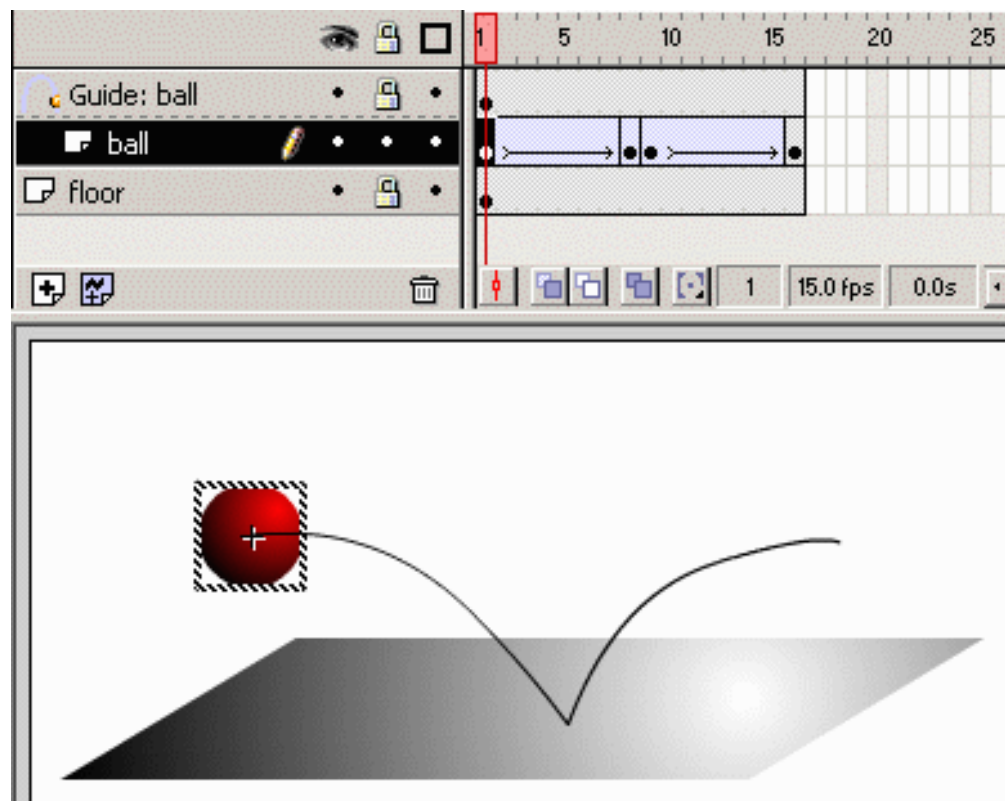
Using Guides

To use the Motion Guide you've created, select the Layer you wish to apply the Guide too and select Insert/Motion Guide. A Guide Layer is placed above the currently selected Layer and the name of the Layer that the Guide is applied to is indented in the Timeline.

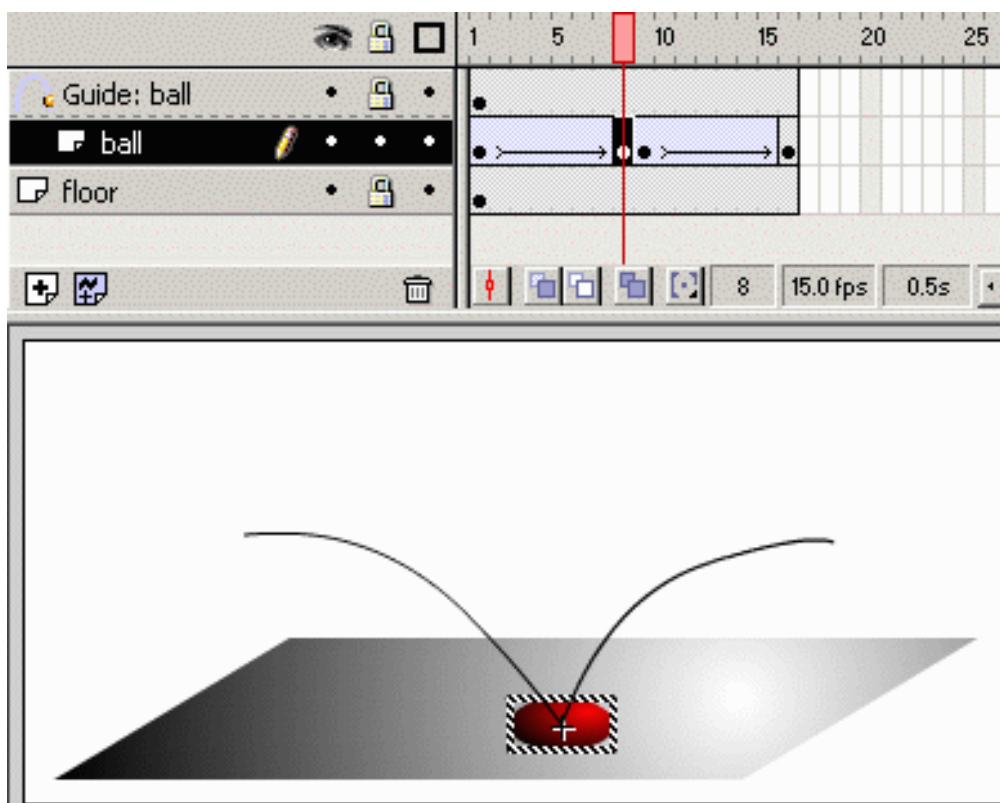


The Guide Layer is added to the Timeline. Note the title of the Guide Layer is Guide: followed by the name of the Layer it is applied to.

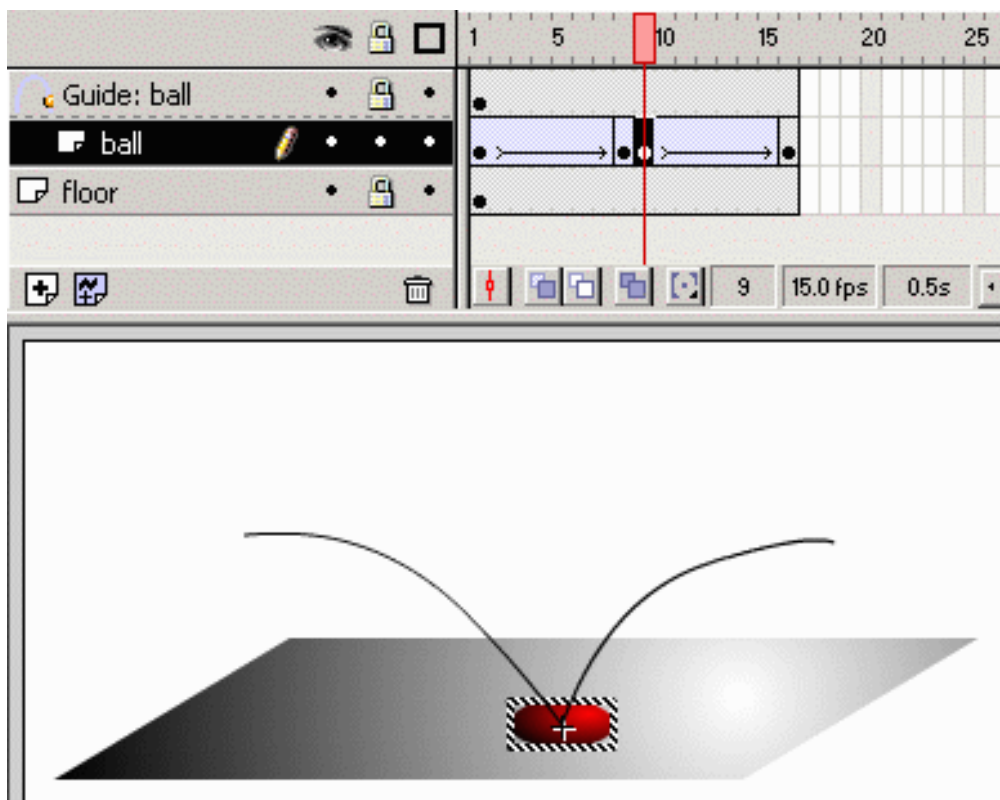
Make sure the new Guide Layer is selected. Drag the Guide Symbol from the Library onto the Stage and position it appropriately. You must break apart the Guide Symbol in order for the Motion Guide to work. Guides must exist on the Shape Level. Once the Guide is broken apart, lock the Guide Layer and select the Layer you're applying the Guide too. Make sure Snap is enabled by selecting View/Snap. Select each key frame in the animation and position the Instance along the Guide. The Instance will snap to the end points along the Guide.



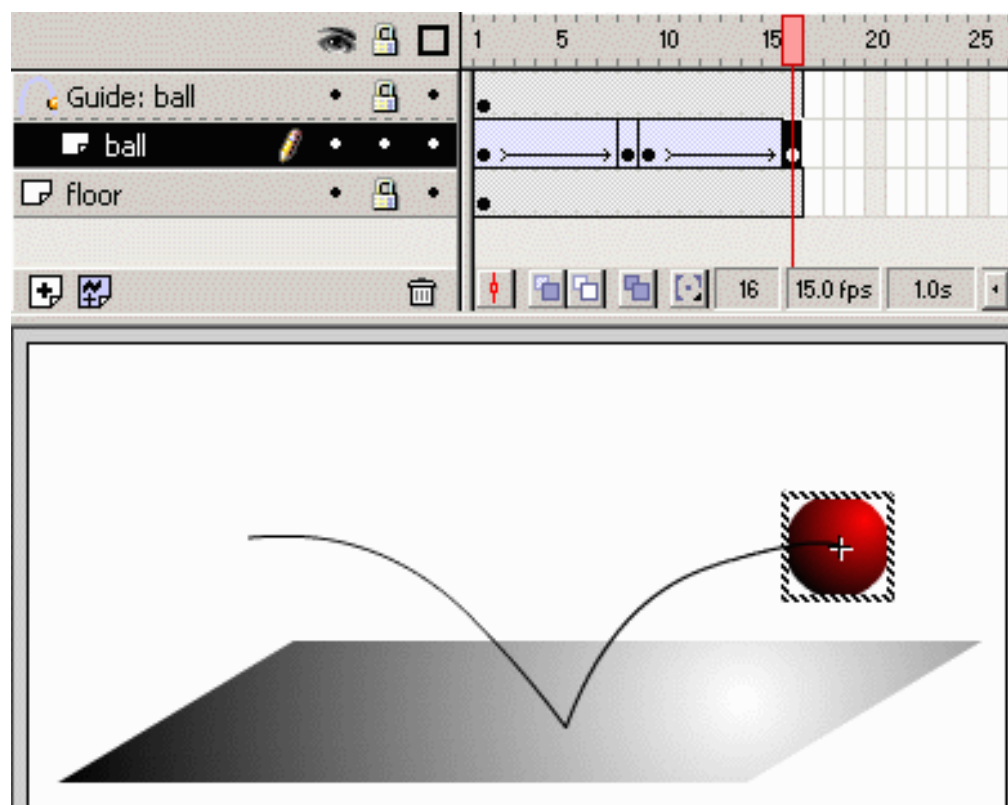
In the first key frame, we position the Instance at the beginning of the Guide.



In the second key frame, we position the Instance at the impact point where the Guide hits the floor.



In the third key frame, we position the Instance at the impact point again, where the Guide hits the floor. This is the second part of the squash sequence.



In the fourth key frame, we position the Instance at the end of the Guide.

Press [ENTER] to view the animation. The Instance should follow the Guide. When you publish the movie, the Guide will be invisible. To preview the published version, press [CTRL] + [ENTER]. You can close the preview by {Clicking} on the Close icon in the title bar or press [CTRL] + [W]. You will be returned to the Flash authoring environment.

If the Instance does not follow the Guide, try to reattach the Instance in each key frame to points on the Guide. If you modify the Guide in any way, you will have to reattach the Instance in each key frame again.

Open the file *Guide.fla* from the *Chapter07* folder and view the animation.

Shape Tweening

Shape tweening involves transforming or morphing one shape into another. You cannot Shape tween Instances or Groups. The shapes you transform must exist on the Shape level. Procedurally, it's a good idea to create your shapes, convert them to Symbols, and break them apart for Shape tweening. That way you always have the original shape as Symbol for use again.

Creating Shape Tweens

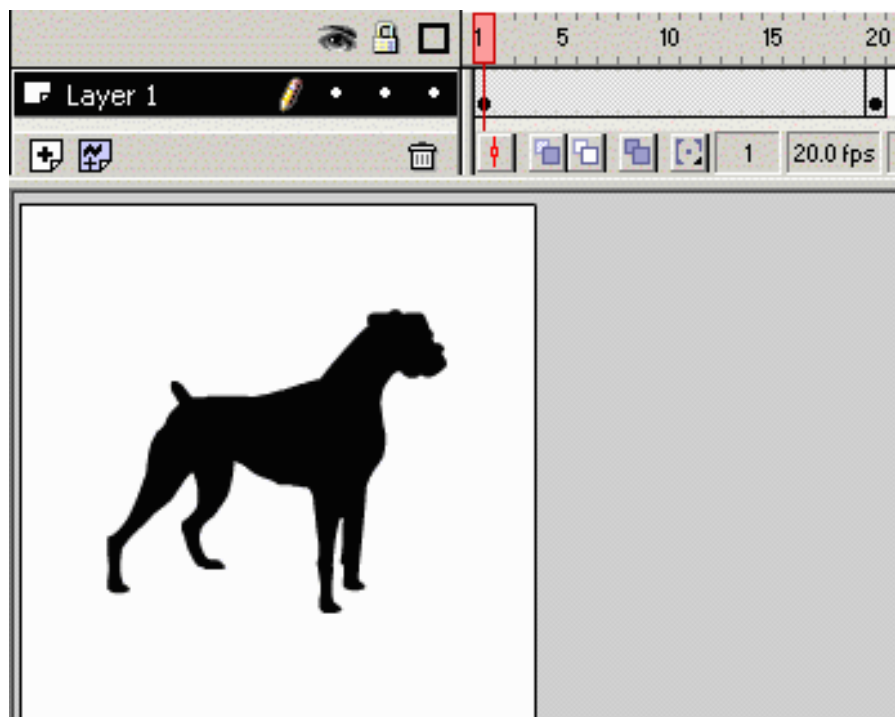
To create a Shape tween, first create the shapes you desire to morph, one into the other. For instance, we're going to transform the shape of a dog into that of a cat.



The dog shape will transform into the cat shape.

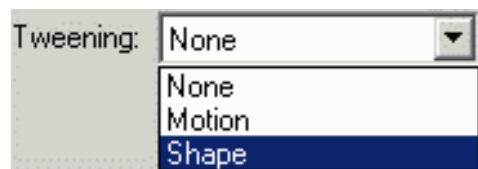
We begin with the dog. We rename our Layer that we'll be placing the Shape tween on too shapes, and place the dog Instance on the Stage, using the alignment controls to center the dog on the Stage. Then we break the Instance apart, ensuring that our shape is in fact, on the Shape Level.

Next we need to create another key frame. Of course, because Flash replicates the previous key frame, the dog shape is present in our new key frame. Unlike Motion tweening, where we animate a single Instance at different points in time, with shape tweening, we animate multiple shapes, or rather, transform one shape into another. We need to delete the dog shape from this key frame and place the cat Instance at this point instead, making sure we break it apart and placing it in the center of the Stage using the Alignment controls.

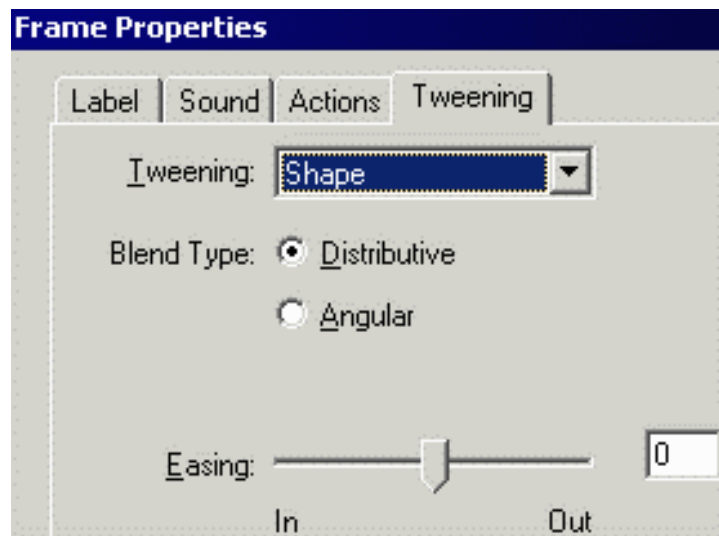


Two key frames, at Frame 1 and Frame 30. The key frame at Frame 1 contains the dog shape. The key frame at Frame 20 contains the cat shape. Although all the shapes are Symbols in the Library, they must be broken apart to enable Shape tweening.

To create the Shape tween, *{Double-Click}* in between key frames just as you do to create a Motion tween. From the Tweening drop-down menu, select Shape.



Shape Tweening options include Blend Type and Easing. Easing controls work precisely as they do for Motion tweening, with Easing In causing the animation to start slowly and gain speed, and Easing Out causing the animation to slow as it ends.



Blend Type

Select from Distributive or Angular. Distributive blending is smoother, and is best used for organic or smooth shapes. Angular blending attempts to preserve the sharp edges and corners of shapes.

For this particular animation, Distributive is appropriate.

When you play the animation, the dog morphs into the cat. A dog and a cat have correlation in their shape, a head, a tail, a body, and four legs. Currently, the head and the two front legs morph together quite well. However, the rear leg of the dog transforms into the tail of the cat, and one of the cat's legs emerges from the stomach area. We have some degree of control over the transformation of one shape to another with Shape Hints.

Open the file Shape.fla from the Chapter07 folder and view the animation.

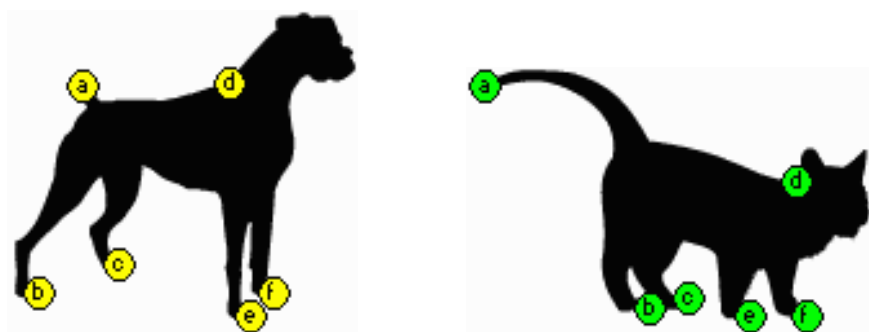
Shape Hints

Shape hints are reference points which you place on each shape. When Flash Shape tweens the shapes, it correlates the Shape hint in one shape with the location of the Shape hint in the other shape. To add Shape hints, select the key frame in which your first shape is placed and select **Modify/Transform/Add Shape Hint**. A Shape hint appears in the middle of the Stage. Each shape hint has a letter assigned to it, beginning with a. As you add more Shape hints, the letters proceed alphabetically.



The first Shape hint.

Reposition the Shape hint by *{Clicking}*, holding, and dragging it to the applicable location. Move to the next key frame and place the same Shape hint on the second shape in the applicable position. Add more Shape hints to further direct the way the shapes transform. You may have to experiment extensively to achieve the desired effect.

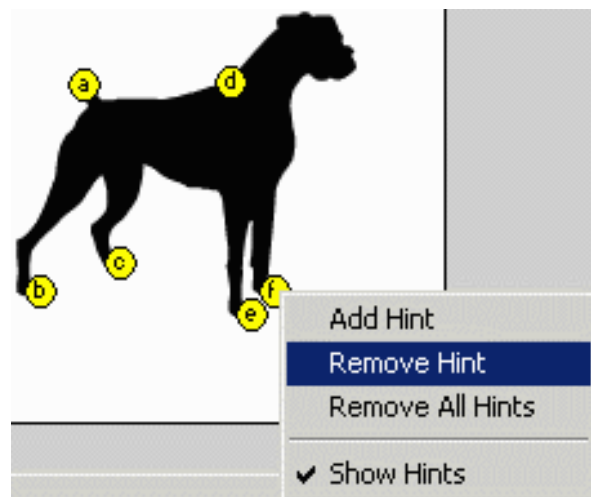


Both Shapes with Shape Hints applied. Shape hints act as a guide in transforming from one shape to another. The a Shape Hint on the Dog will correlate with the a Shape Hint on the cat. The b Shape Hint on the Dog's rear leg will correlate with the b Shape Hint on the Cat's rear leg and so forth.

Open the file `Hint.fla` from the `Chapter07` folder and view the animation.

Removing Shape Hints

To remove Shape hints, *{Click}* on the Shape hint you want to remove and *{Right-Click}*. Select Remove Hint from the context-sensitive menu.

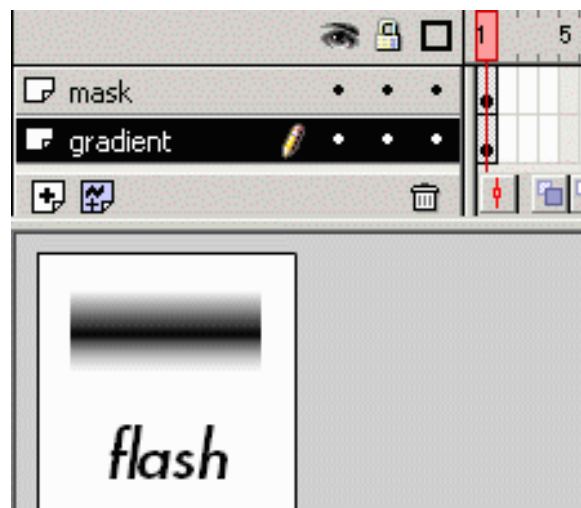


Use a context-sensitive menu to remove a single Shape Hint.

To remove all Shape hints, you can use the context-sensitive menu and select Remove All Hints or select Modify/Transform/Remove All Hints.

Using Mask Layers in Animation

In addition to Guide Layers, you can make Layers in Flash act as masks. Masks hide the content of the Layer they are on top of in the Timeline. The Instance that you place on a Mask Layer acts as the hole in the mask. Any content on Layers affected by the Mask Layer will only be revealed if the Instance on the Mask Layer passes over it.

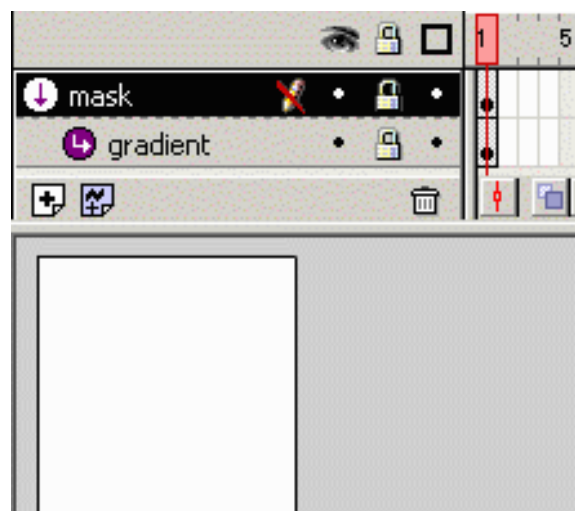


Two Layers. One with a gradient and one with text flash. The text is on the Layer we've called mask.

To designate a Layer as a mask, {Right-Click} the Layer and select Mask from the context-sensitive menu.

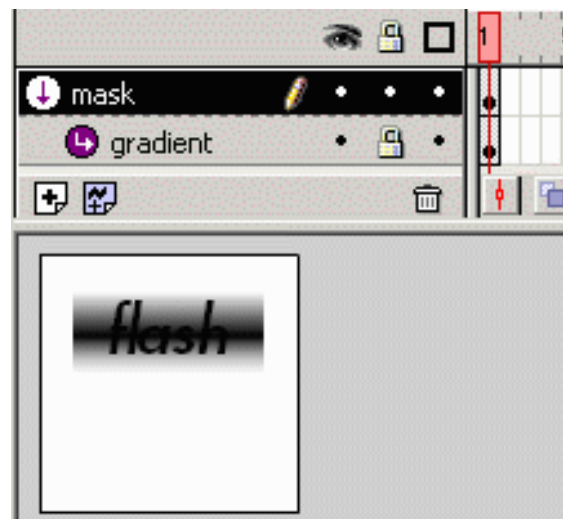


The Layer will mask the Layer directly below it.

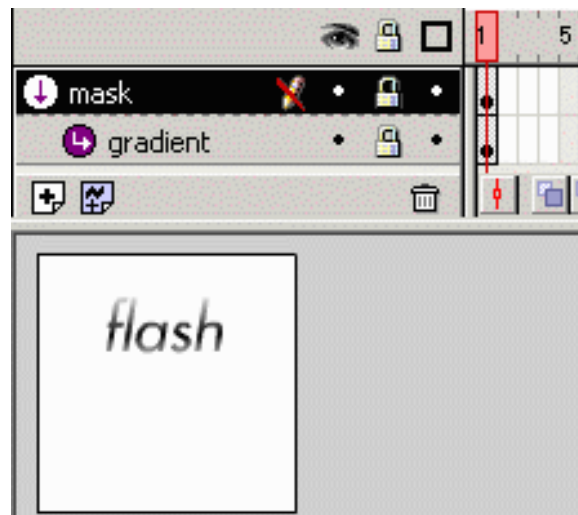


Because the text flash on the mask Layer is not over the gradient Instance on the gradient Layer, the gradient is no longer visible. You have to place the Instance on the Mask Layer over the Instance below it to make it visible.

Flash will indent the Layer below the Layer Mask and lock both Layers. To edit content, unlock the Mask Layer. When a Mask Layer is unlocked, it temporarily disables masking. Lock the Layer again to enable masking.

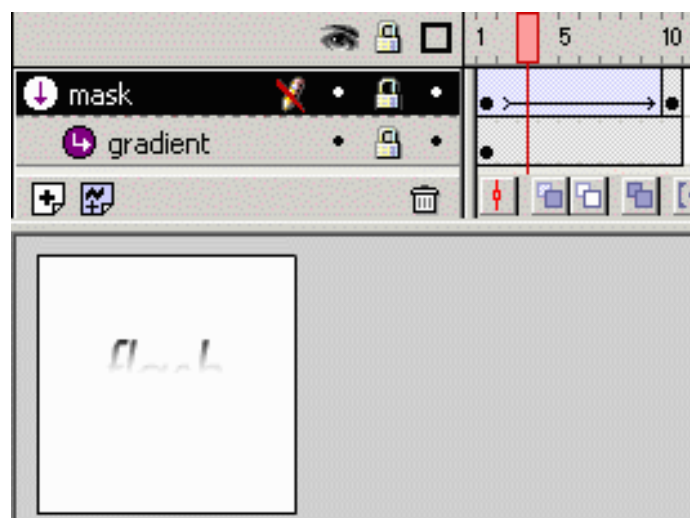


Unlocking the Mask Layer temporarily disables the mask. Now we can edit the Instance on the Mask Layer. We've placed it over the gradient Instance.

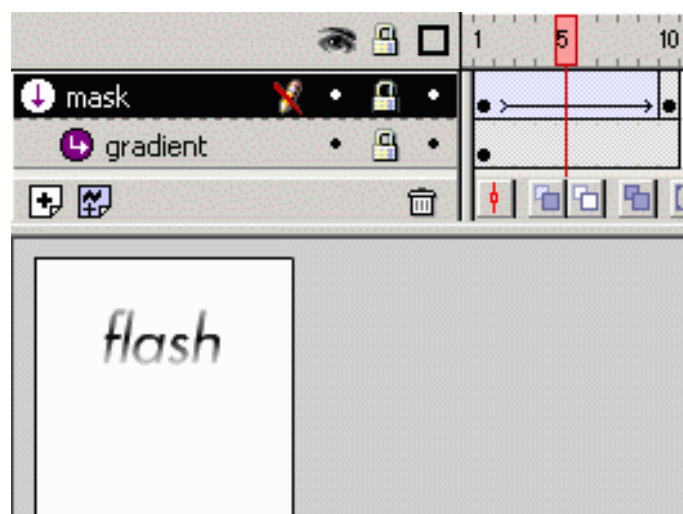


Locking the Mask Layer enables the mask. The text on the Mask Layer acts as a hole in the mask through which we see the gradient.

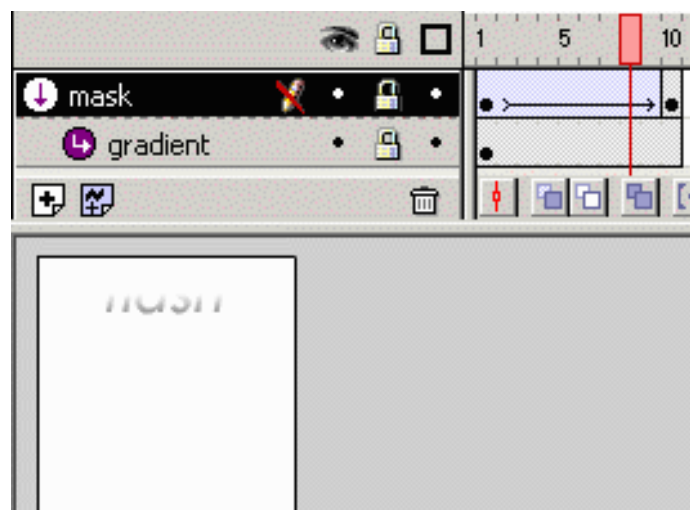
You can animate content on Mask Layers and content on Layers which are masked. In this case, we animate the flash Instance passing over the gradient. Instances on Mask Layers can only be solid in color. As the flash Instance passes over the gradient, it appears to fade in to view, solidify, and fade out of view as it moves up the screen.



The flash Instance begins to pass over the gradient below.



The flash Instance passes over the middle part of the gradient.

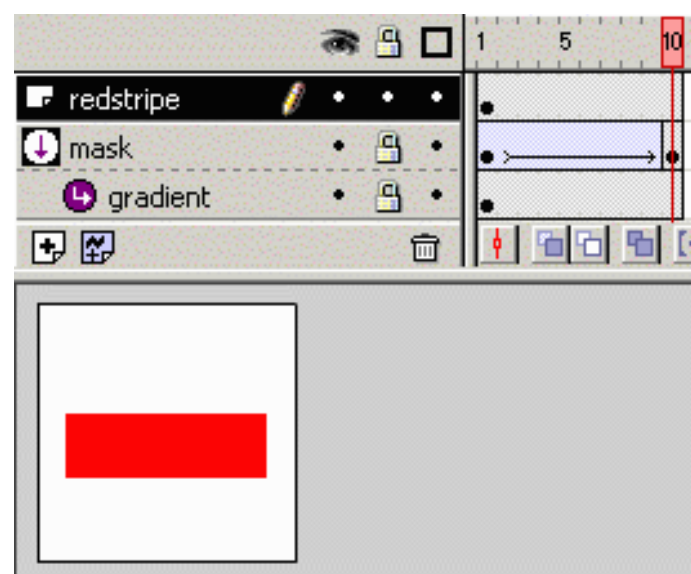


The flash Instance passes over the top of the gradient.

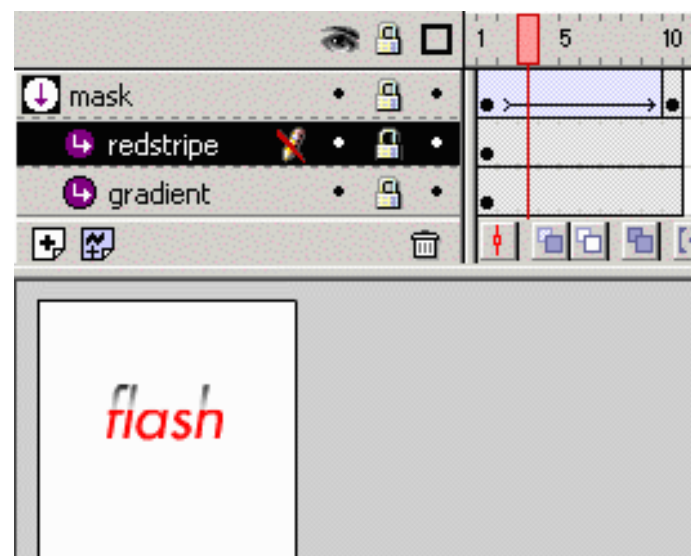
Open the file Mask.fla from the Chapter07 folder and view the animation.

Masking Multiple Layers

By default, when you designate a Layer as a Mask, it masks the Layer directly below it in the Timeline. To mask additional Layers, simply drag and drop a Layer in the Timeline directly below the Mask Layer.



A Layer called redstripe with a red box Instance on it.



By dragging the Layer redstripe below the Mask Layer, it becomes masked as well.

Exercises

Exercise # 1

1. Use Motion Tweening to create a bouncing ball animation as described in the chapter.
2. Remember to place each Instance its own Layer.
3. Try out the Easing Controls for Motion Tweens.

Exercise # 2

1. Add a Motion Guide to your bouncing ball animation.
2. Remember that the Guide cannot be a Symbol. It must exist on the Shape Level.

Exercise # 3

1. Shape tween two shapes, a circle and a square.
2. Open Shape.fla from the Chapter07 folder and add Shape Hints to control the tweening.

Exercise # 4

1. Use a Mask in an animation.
2. Make your name a Symbol.
3. Reveal your name by animating a block passing over.
4. Make sure to make the Layer that the block is animated on is designated as a Mask.

Review Questions

1. How many Instances can you Motion Tween on a Layer?
2. What effect does easing in or out have on a Motion Tweened Instance?
3. Describe the process of adding a Motion Guide to an animation.
4. Can Instances be used in Shape Tweening?
5. When designating a Layer as a Mask, does the Instance on the Mask Layer cover or reveal the contents of the Layer it is masking?
6. How do you add additional Layers to Mask Layer?

Summary

As a result of this chapter, you should be able to

- Use Motion Tweening
- Use Guide Layers in Animation
- Use Shape Tweening
- Use Mask Layers in Animation

CHAPTER 8

Movie Clips



Movie Clips are animated Symbols. Each Symbol has its own Timeline on which you can animate Instances from other Symbols. When you place a Movie Clip Instance on the Stage, the animation can play in a single frame on the Timeline. Movie Clips are an important element in creating animations, buttons, and interactivity.

OBJECTIVES

Upon completing this section, you should be able to:

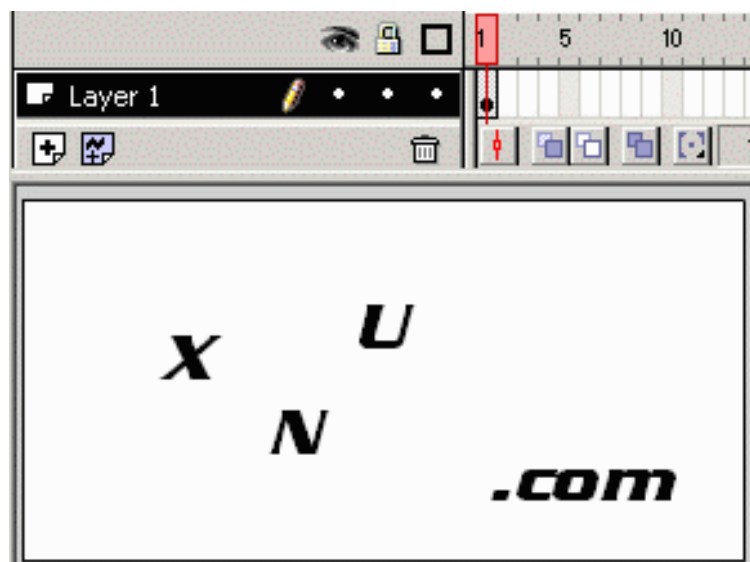
1. Create Movie Clips
2. Use Multiple Timelines
3. Understand Parent and Child Objects
4. Use Movie Clips on the Stage

Using Movie Clips for Animation

Instances of Symbols can be used repeatedly, so building most of your animation with Movie Clips gives you the freedom to use the animations you create again and again. Because Movie Clips are Instances, you can apply any of the regular Instance properties to entire animations, such as scaling, rotating, and changing transparency. Using Movie Clips requires more planning on your part, but it also gives you increased flexibility.

Creating Movie Clips

To create Movie Clips, first create all the graphic or static Symbols that will be Instances in your animation. Simply make sure that a Layer is active in the Timeline and build each shape on the Stage, then convert each shape into Symbols and delete them off the Stage. We will only be placing the Movie Clip on the Stage.



4 Shapes (text on the Overlay level) on Stage.

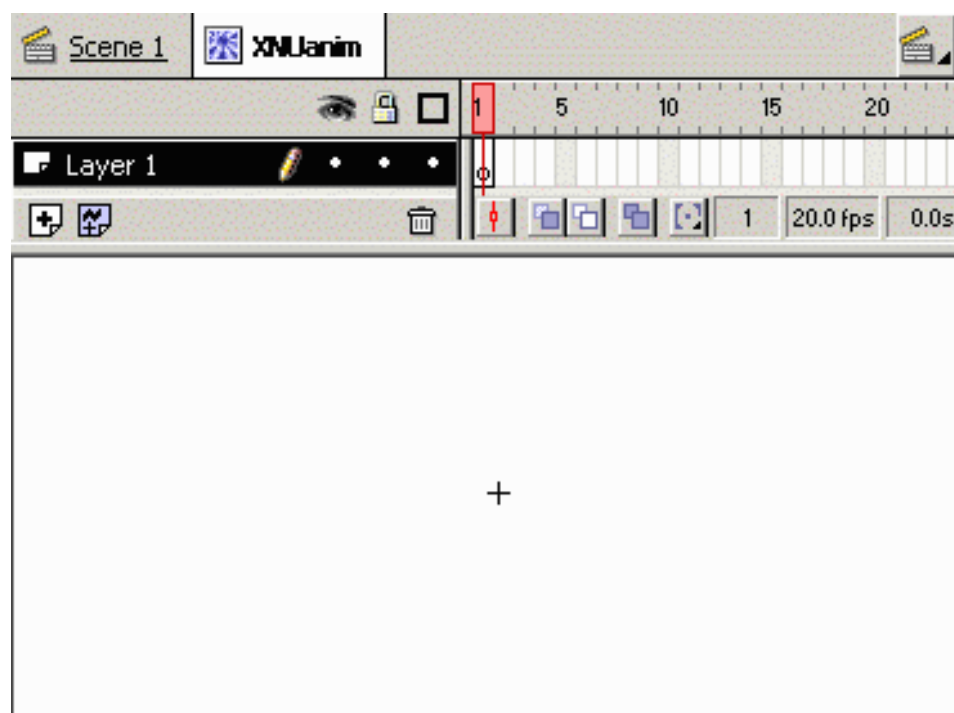
Convert them to Symbols and use the Instances in the animation you build.

Next we need to create the Movie Clip. {Click} the New Symbol button  at the bottom of the Library. Name the Symbol appropriately and make sure to select Movie Clip for Behavior.



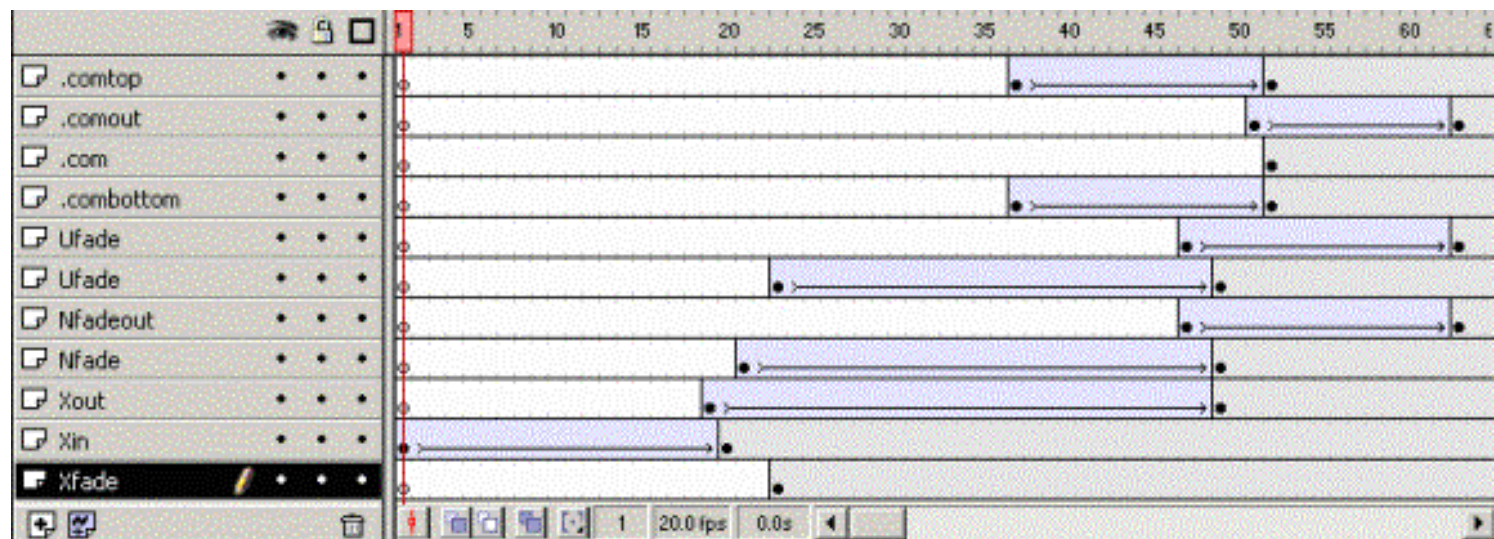
Select Movie Clip under Behavior when making animated Symbols.

After you {Click} OK you are instantly placed within the Symbol's Timeline. Recall that you can tell you are on the Symbol's Timeline and Stage because a crosshair appears in the center of the Stage and the name of the Symbol appears as a highlighted Tab at the top of the Timeline.



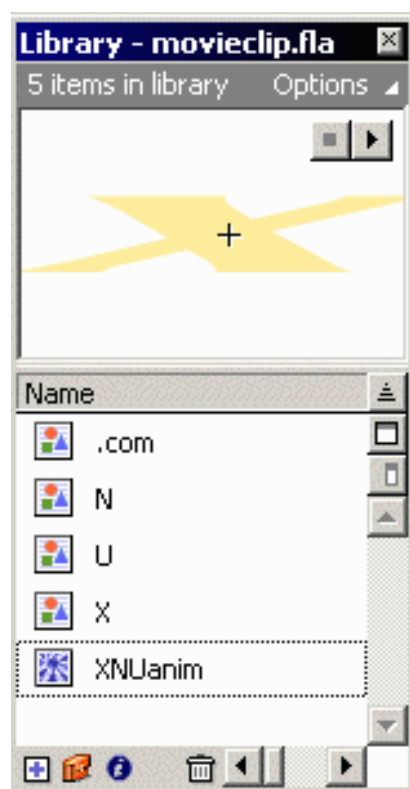
Having created the Symbol, you are placed within the Symbol. Proceed to build the animation.

Next we can build the animation just as we build animations on the main Timeline and Stage, by dragging Instances of Symbols from the Library and animating them on separate Layers.

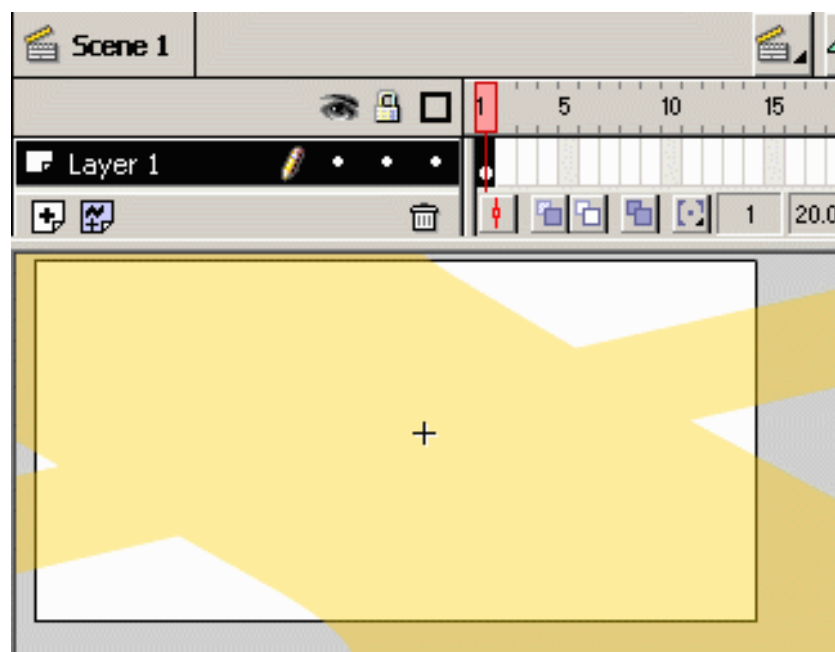


The animation is built inside the Movie Clip Symbol's Timeline with Instances of the static Symbols created earlier.

Once you've completed creating the animation in the Movie Clip Symbol, exit to the main Stage and drag an Instance of the Symbol onto the Stage.

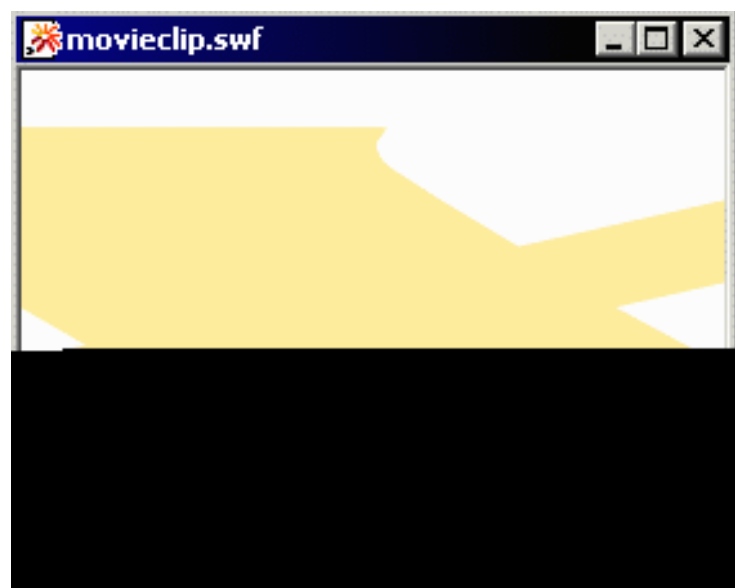


Movie Clips are instantly recognizable in the Library, by the Movie Clip icon .

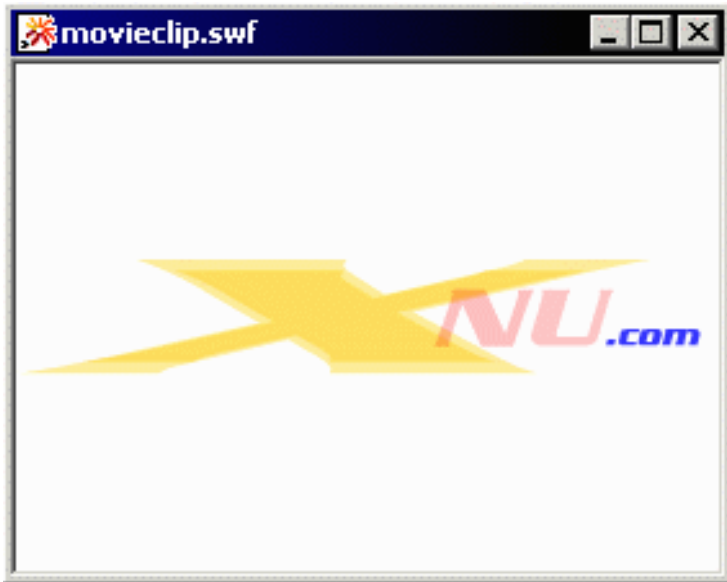


The Movie Clip placed on the first key frame in Layer 1.

You cannot view the Movie Clip in the production window. To view it, press [CTRL] + [ENTER] to see the Shocked version. You may have to adjust the size and position of the Movie Clip accordingly. The Movie Clip's selection box is sized to the largest Instance in the animation, making it difficult to know how the entire animation will appear on the Stage. In this case, the X of XNU starts at a very large size and ends well positioned on Stage. The X is so large at the beginning of the Movie Clip that it actually spills off the Stage. This is the effect we want.



In the Shockwave Movie, the X begins by spilling of the edge of the screen.



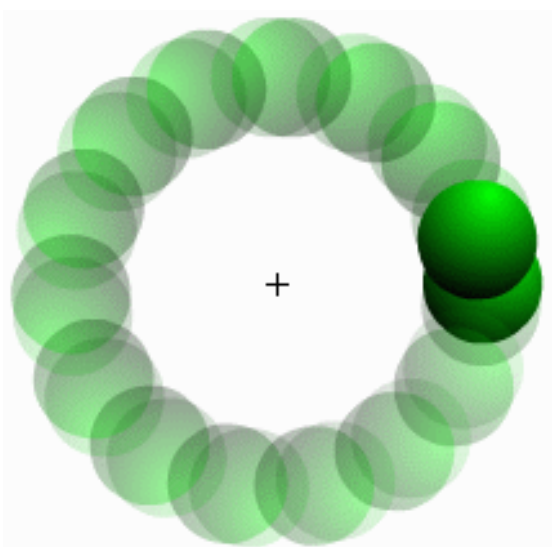
At the end, the animation comes to a stop and the full logo is on the Stage.

Open the file MovieClip.fla from the Chapter07 folder and view the animation.

Using Multiple Timelines

It's very important to understand how multiple Timelines operate in Flash. Each Symbol has its own Timeline. Symbols with Movie Clip as Behavior use their own Timeline to animate Instances of other Symbols. Furthermore, you can embed Movie Clips within Movie Clips, and if you're a careful planner, you can embed Movie Clips within Movie Clips, within Movie Clips! The key to effective and dynamic animation in Flash is combination. Understanding how each element works is essential, and the combination of each element helps you create great animations and streamlines the production process.

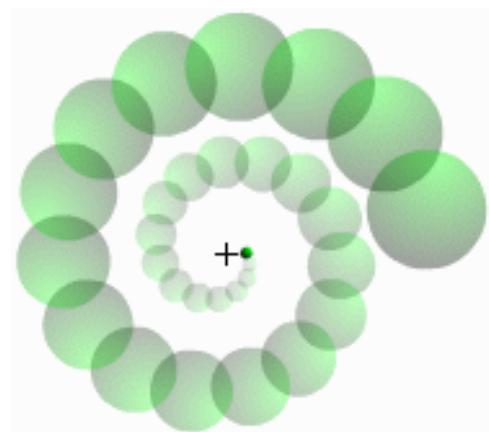
For example, in a particular project, we have two animation objectives among many. First, we need to have a ball circle around a point, as if in orbit around an invisible planet.



The ball circles or orbits around the center of the Stage. Onion Skinning is enabled to see where

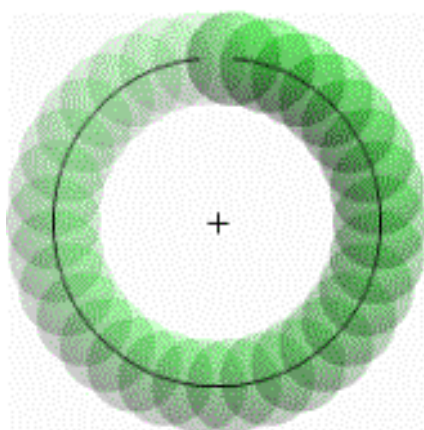
the ball is in each frame.

Second, we need to create an animation of a ball circling around a point in the distance and getting closer, as if following a helical or spiral path.

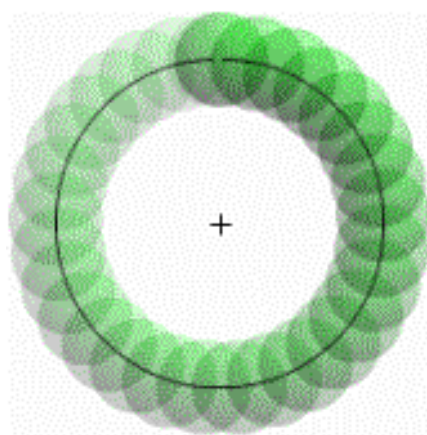


The ball spins out toward the viewer. Onion Skinning is enabled to see where the ball is in each frame.

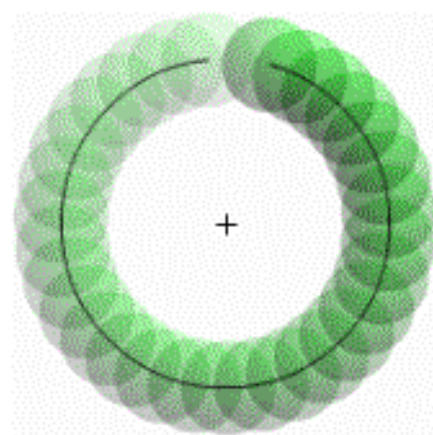
Our first step is to think about how we might achieve our animation objectives. Our first thought might be to use a Motion Guide in both instances. However, there is an immediate problem with the circular animation. Motion Guides require an individual point for each key frame. A circular path has only one point. We would have to create a gap in the circular path. Furthermore, the gap would have to be precisely the right size to maintain a perfect 360 degree rotation, so that when the animation loops, the space between the beginning and end points equals the space between each point in each frame, making the circular motion flawless. If the gap is spaced too closely, the ball will appear to slow or pause very slightly at the top of its path. If the gap is too large, the ball will appear to jump or speed up at the top of the path. It will take considerable work, or luck, to be precise.



Precisely sized Gap



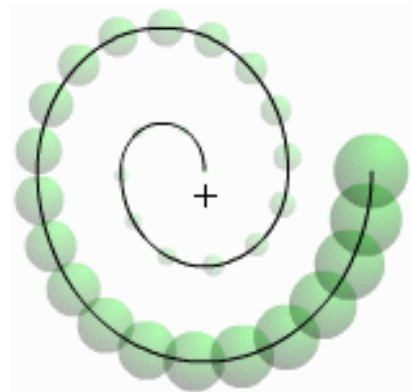
Small Gap



Large Gap

Using a guide for this type of motion may prove difficult. It is a valid solution however.

To create our spiral animation, we could create a spiral path. It would be difficult to create a spiral path using Flash's Drawing Tools, we would need to create it in a vector drawing program, such as Macromedia's Freehand, and import it into Flash. Once we have our spiral path, we would attach a ball Symbol to each end, reducing the scale of the ball Instance at the first key frame and increase the scale of the ball Instance at the last key frame. As the ball follows the path, it will increase in scale, creating the illusion that it is moving from a distance and coming closer to the viewer.

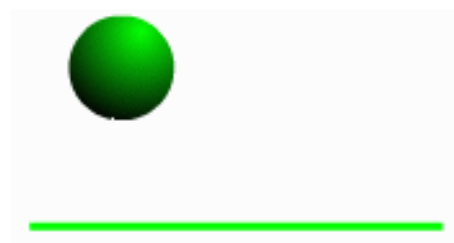


The Movie Clip containing the spiral path and the ball Instance following it.

Using Parent and Child Objects

There is another method which is less time consuming, and doesn't involve using Guides at all. Many animation effects use a specialized technique involving attaching a child object to a parent object. We animate the parent object and the child object is attached to the parent and follows it. Once again, the key is taking particular and fundamental elements of Flash, and combining them to achieve results. As you use Flash, you're experience increases, and as you experiment, you'll find new and interesting ways to achieve particular results.

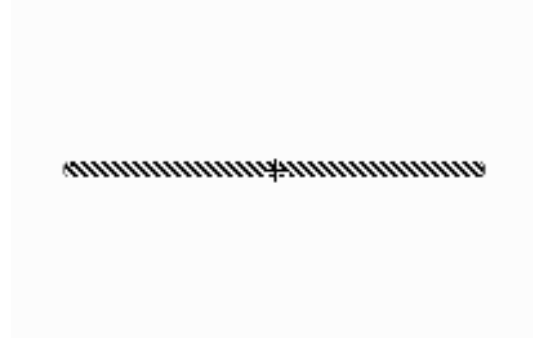
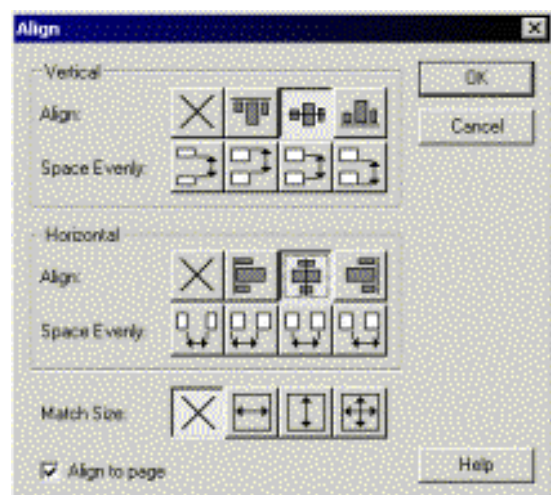
Our first step is to create the static Symbols we need for our animations. We need a ball shape, which will do for both animations. Because we're not using paths, we need one other shape, a rectangle, which will be our parent object. We will animate the parent object, and attach the ball Instance to the parent. Parent objects are made invisible to the viewer, but act as guides for visible objects, such as the ball.



The ball and rectangle Symbols.
The rectangle will be a Parent.

Once we have built our static Symbols, (named ball and parent) we need to build our Movie Clips. We'll start by building the circular animation, where the ball moves in a circular motion around a point (named circular).

How do we do this without a path? We use the Parent. Within our Movie Clip, we need to create one more Symbol, which will be a combination of the Parent and the ball. First we place a parent Instance in the center of the Stage using the Align Tool (Align Center, vertical and horizontal, align to page).



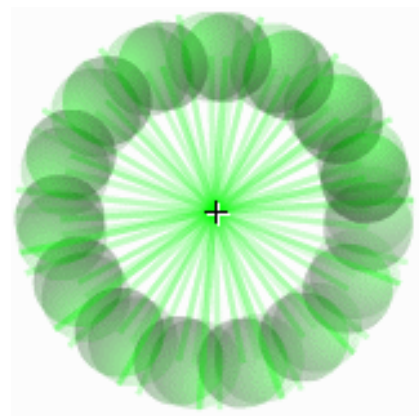
Using the Alignment Tool, the Parent Instance is centered on the Stage.

We center the Parent, so that its Center Point is aligned with the Center Point of the Movie Clip Symbol. Why do we need to do this? This permits us to rotate the Parent along the center axis, making for a perfect and balanced 360 degrees of rotation. Next we need to convert the parent Instance into a new Symbol. This new Symbol will be a combination of the Parent and the ball (named parentandball). The moment you create a new Symbol on the Stage (the main Stage or Symbol Stage), you are instantly working with an Instance of the Symbol. Next we need to animate it. We Motion Tween the Instance, using a Clockwise rotation with a value of 1, 30 frames in length. Our animation is done. Next, we must attach the ball Instance to the end of the Parent. To do so, we edit parentandball, the Symbol which we created in the Movie Clip circular, and with Snap enabled, we attach the ball Instance to the edge of the Parent. Now the circular Movie Clip features a rotating Parent, with the ball on the edge of the Parent.



The Parent and Ball Instance combined.

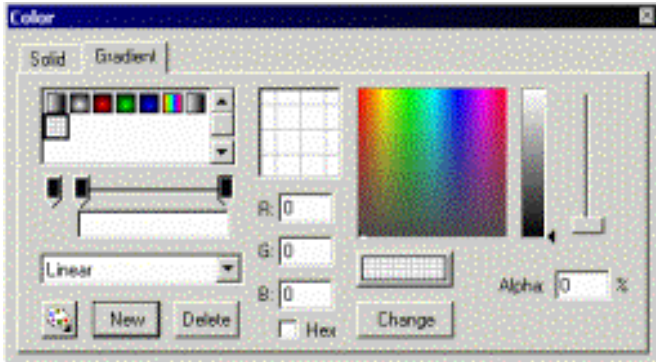
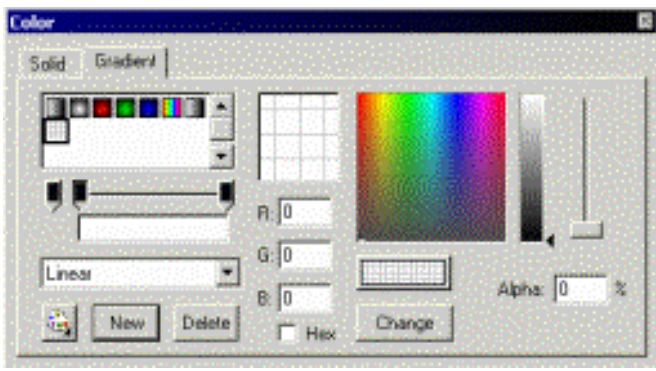
The ball, attached to the Parent, circles around a point. However, there is one problem. Flash's Motion Tween Rotation controls automatically make the first key frame identical to the last key frame. In other words, the Instance begins at 0 degrees, is automatically rotated a full 360 degrees, ending at a position of 0 degrees again. This will cause a slight pause in the animation. In this case, the animation is 30 frames in length, with key frames at Frame 1 and Frame 30. To correct this problem, we need to select Frame 29 on the Timeline, insert a key frame, clear the keyframe in Frame 30, and drag the key frame in Frame 29 to Frame 30. Now the circular motion is perfect.



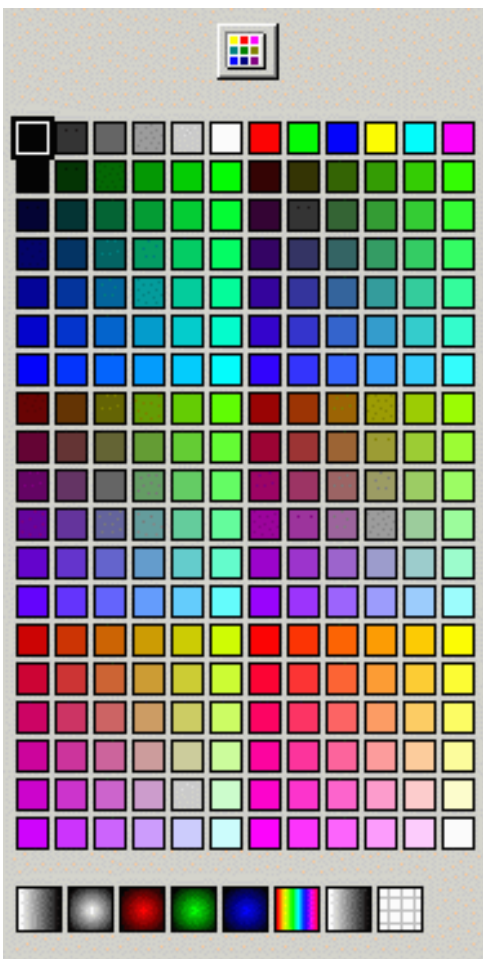
The circular Movie Clip, animated across 30 frames.

Open the file Parent.fla from the Chapter08 folder and view the animation.

At this point, the Parent is still visible. To make it invisible, we have two options. One, we could edit the parentandball Symbol, select the parent Instance, and change its alpha value from 100% to 0%. We could not do this to the parent Symbol because the rectangle that makes up the Parent is still a Shape within the parent Symbol. (Recall that when you convert a Shape into a Symbol, within the Symbol's Timeline, the Shape remains a Shape). In the parentandball Symbol however, we have an Instance of the parent Symbol within it, and hence we can apply Instance properties such as Alpha. Our other option is to modify the Shape within the parent Symbol itself. We could fill it with a transparent color or gradient. Its up to you to decide which method to use. We recommend using the Gradient method because Gradients are placed at the bottom of the Color Palette, making them easier to select. Its comes down to personal preference. We could make a new Linear Gradient, with Black Markers at both ends, each with an Alpha value of 0.



A Custom Linear Gradient with both Markers set to an Alpha of 0 creates a transparent Fill color.



The transparent gradient appears amongst the gradients in the color palette.

This is the preferred method because if we use an Instance of the parent Symbol within many other Symbols, we would have to edit each Symbol in turn, and change the Alpha value of the parent Instance in each Symbol. Using the transparent gradient method, every Instance of the parent Symbol will be transparent. Generally, we keep Parents visible until we've completed our production as it's easier to edit visible objects. However, you can also work in Layers as Outlines mode as well.

Open the file `Transparent.fla` from the `Chapter08` folder and view the animation.

Finally, we need to create one more Movie Clip for the spiral animation (named spiral). We have two options. One, we could duplicate the circular Symbol and reduce the scale of the parent and ball Instance in the first key frame. As the animation proceeds, the Motion Tween will increase the scale of the Instance, making it appear to move closer as from a distance. Or we could create a new Symbol and use an Instance of circular. In the first key frame, the Instance would be reduced in scale, and in the last key frame, the Instance would be increased in scale, creating the illusion that the ball is moving closer to us on a spiral path.

Open the file `Spiral.fla` from the `Chapter08` folder and view the animation.

To Review:

Create Static Symbols

We create two shapes on the main Timeline, one a ball, and one a rectangle. We convert each shape into a Symbol, one called ball and one called parent. The Behavior for each Symbol is Graphic.



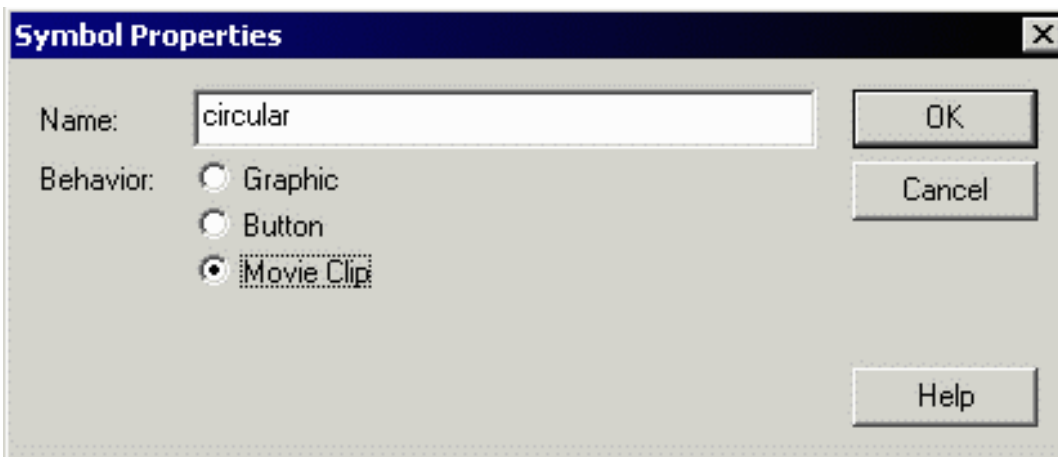
The ball shape.  ball The Symbol in the Library.

 The rectangle shape.

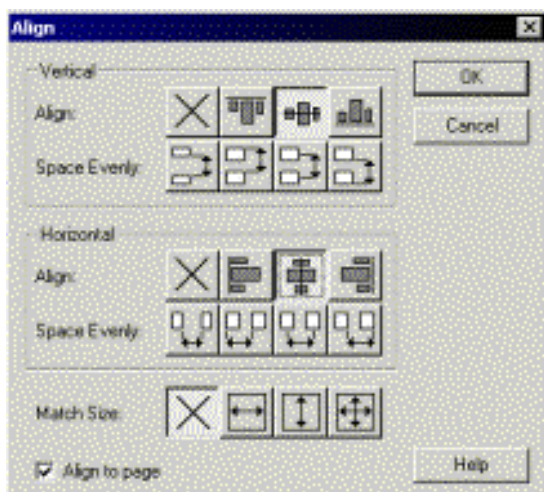
 parent The Symbol in the Library.

Create First Movie Clip Symbol

Next we create the circular animation, where the ball appears to circle around a point, as if in orbit around an invisible planet. We {Click} the New Symbol button in the Library, call the Symbol circular, and designate Behavior as Movie Clip.

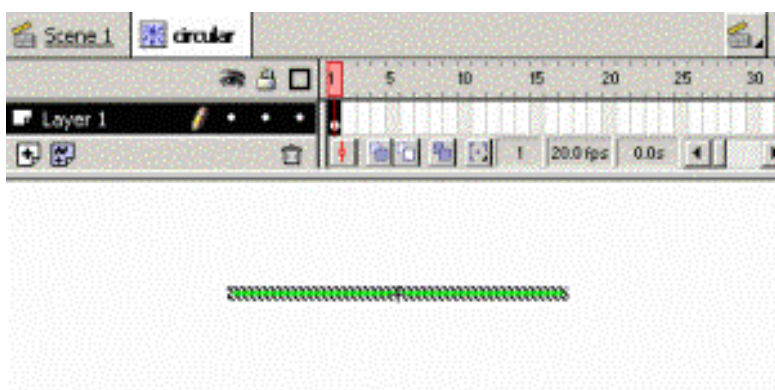


Next we drag an Instance of parent on to the Stage, using the Align Tool to center it relative to the Stage.



Make sure to **{Click}** Align to page in the Align dialog-box to center the Instance in the center of the Stage.

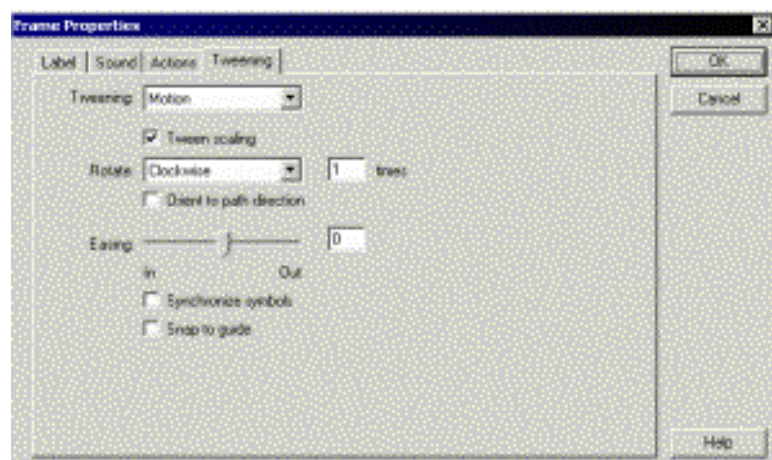
Next we select the Instance and convert it into a new Symbol with a Behavior of Graphic. Why? Because we might want to use the parent Instance in another animation with a different object. For each object we are going to constrain with a Constrainer, we need to create a new Symbol. We'll call the new Symbol parentandball. The moment we convert the parent Instance into a new Symbol, we are working with the Instance of the new Symbol, in this parentandball on the Stage. We are still within a Symbol, the Movie Clip circular.



Editing the Movie Clip circular. We've dragged the parent Instance on to the Stage and centered it relative to the Stage. Then we've converted this Instance into a new Symbol, called parentandball.

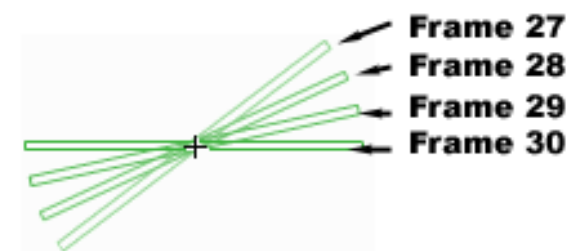
Now we are working with the new Instance parent and ball.

Now we can animate the Instance. We create a new key frame in Frame 30 and Motion Tween the Instance. We select Clockwise from the Rotate control and input a value of 1 into the times field. This will automatically rotate the Instance a full 360 degrees within 30 frames.



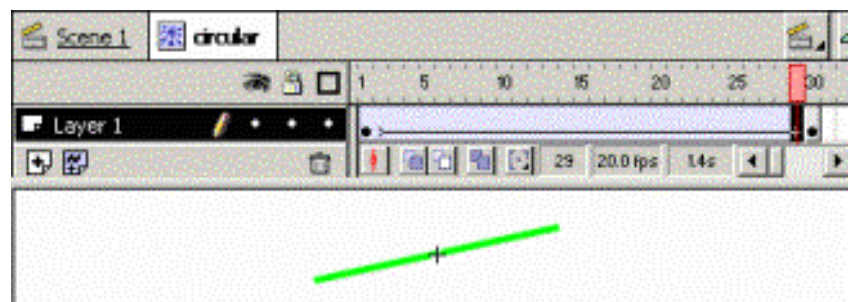
Setting Motion Tweening to a 360 degree Clockwise rotation, 1 time within 30 frames.

Because of the way automatic Rotation works in Motion Tweening, the position of the Instance is identical in both key frames. This is a problem because as the animation loops, Frames 2 through 29 are unique in regards to the position of the Instance. Frames 1 and 30 are duplicates, therefore, each time the animation finishes its cycle and begins again, the Instance will be in the same position for 2 frames, causing a slight pause. We need to eliminate the duplicate key frame.

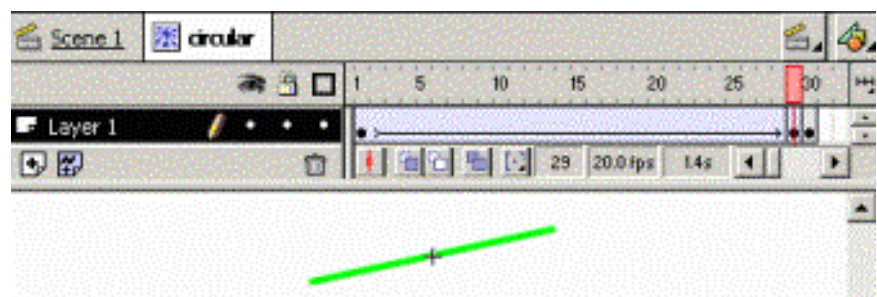


Frames 27 through 30 (onion skinned). Notice there is an even space between the position of the Instance in each Frame. In Frame 30, the Instance is in the exact position as in Frame 1. We need to make Frame 29 the second key frame so when the animation loops, the Instance moves from its unique position in Frame 29, to the unique position in Frame 1.

To eliminate the last key frame, first we select Frame 29 in the Timeline and insert a key frame. When a Motion Tween already exists in a Layer, and you add a key frame, it does not duplicate the previous key frame, instead, it creates a key frame with information based on the unique position and state of the Instance within the Motion Tween.

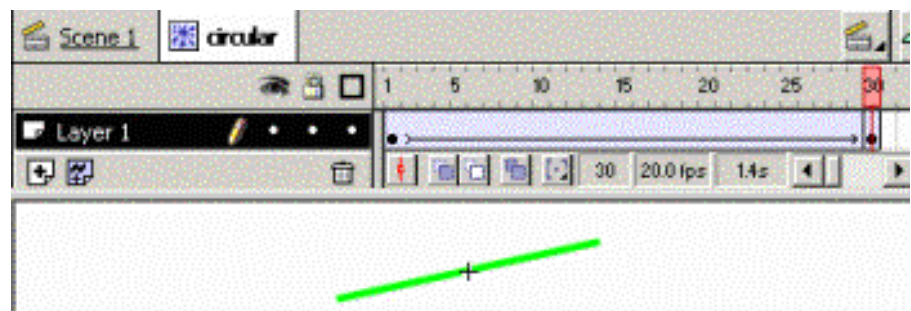


The Marker at Frame 29. Notice the position of the Instance.



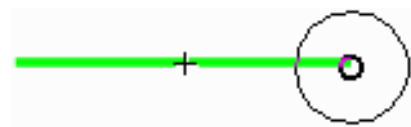
When you insert a key frame at Frame 29, into an existing Motion Tween, the position of the Instance remains.

Next we simply delete Frame 30 from the Timeline by selecting it and selecting Insert/Delete Frame or press [SHIFT] + [F5]. Then we drag the keyframe at Frame 29 to Frame 30. Our animation is complete.

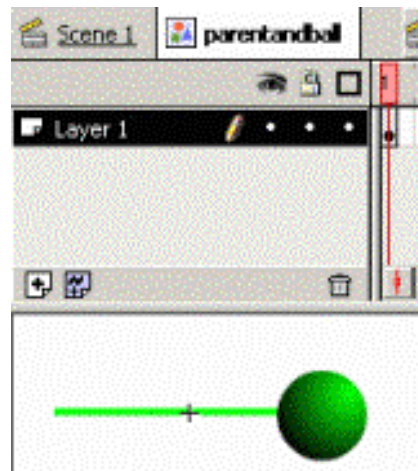


The new key frame at Frame 29 moved to Frame 30. The rotation of the Parent is now seamless.

Now we need to add the ball Instance to the parentandball Symbol. We edit the parentandball Symbol and with Snap enabled (select View/Snap) position the ball Instance at the end of the Parent.

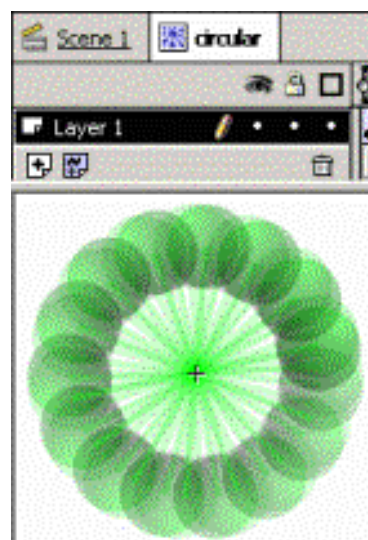


With Snap enabled, as you drag the ball Instance to the edge of the Parent, it will snap to the edge of the Parent.



The ball Instance is attached to the edge of the Parent.

Having modified the parentandball Symbol, the Instance of that Symbol, embedded in the circular Symbol, now features the ball at the edge of the Parent. The ball appears to be circling around a point.



Viewed with Onion Skinning enable, the ball, attached to the Parent, appears to circle around a point. Technically, the Parent is animated, and the ball, attached to the Parent, follows it.

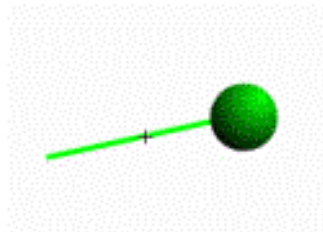
Create the Second Movie Clip Symbol

Finally, we need to create the spiral animation in a Movie Clip. There are two methods we can take. First, we can duplicate the existing circular Movie Clip, rename it spiral and modify the duplicate. Second, we can create a new Movie Clip, and use an Instance of the spiral Symbol within it.

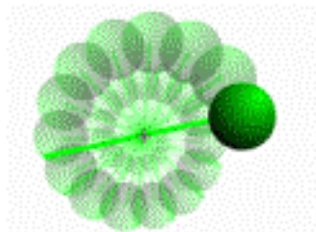
Using method one, we duplicate the circular Movie Clip, rename it spiral and edit the duplicate Symbol. In key frame 1, we reduce the scale of the parentandball Instance by 20%. As the animation proceeds, in addition to the rotation, the scale of the Instance is animated from smaller in scale to larger in scale, creating the illusion of an object moving closer to us from a distance.



The Instance in Frame 1.



The Instance in Frame 30.



The Instance in Frames 1 – 30.

Using the second method, we create a new Movie Clip, name it spiral and drag an Instance of circular onto the Stage, centering relative to the Stage with the Align Tool. You might notice that the Instance's center point is not precisely aligned with the center of Stage. This is not a concern however, because we will not be rotating this Instance, the Instance, because it is the circular Movie Clip, rotates itself. We only need to Motion Tween the scale of the Instance. We create a second key frame at Frame 30. Then we reduce the scale of the Instance in the first key frame and Motion Tween between them. The illusion of an object coming closer from a distance is created.



The Instance in Frame 1.



The Instance in Frame 30.

When you play this Movie Clip in Flash, you'll only be able to see the change in scale. Recall that to fully view Movie Clips, you must place them on the Main Stage and view the Shockwave version of the movie.

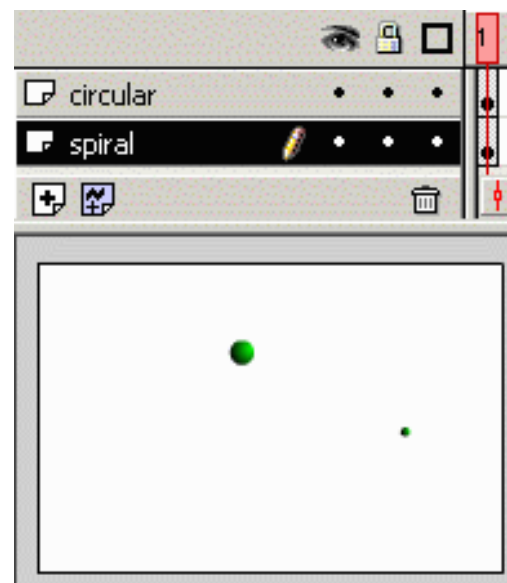
Which is the preferred method? Both methods work well, however, the second method is more flexible because we can decrease or increase the number of times the ball circles around a point. For instance, we know that the parentandball Symbol contains an animation 30 frames in length. Within the range of 30 frames, the Instance rotates 360 degrees once. If we scale the circular Instance across 90 frames, the Instance will circle 3 times as much as scaling it across 30 frames, without increasing the time it takes for the ball to circle around a point. If you extend the range of frames using the first method, it would prolong the time it takes for the circle to make one complete

rotation. You would have to increase the value of the times field in the Motion Tween.

Finally, we need to make the Parent invisible. We edit the parent Symbol and fill the rectangular shape with a custom transparent gradient or custom transparent color. Because there are Instances of the parent Symbol within both the parentandball, circular, and spiral Symbols, the Parent is made instantly invisible.

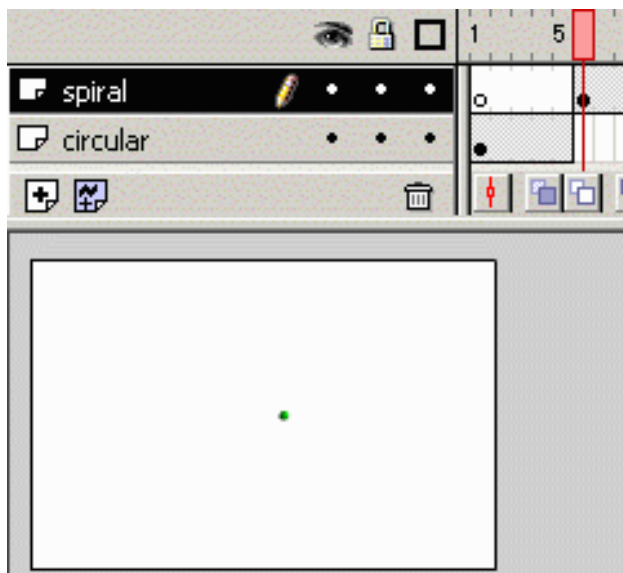
Using Movie Clips on the Main Stage

With multiple Timelines in use, how is timing effected when you place Movie Clips on the Main Stage? Movie Clips will loop within one frame on the Main Timeline. In other words, if the Main Timeline consist of a range of 1 frame, Movie Clips placed in key frames will play through the full range of frames within the Movie Clip.



Two Movie Clips on the Main Timeline. With only a range of one frame, the Movie Clips play through the range of frames within their own Timelines.

The moment you extend the range of frames on the Main Stage, to play through the entire range of any Movie Clips you place on the Stage, you must extend the range of frames on the Main Timeline to match the range of frames within the Movie Clip's Timeline. For instance, if our circular Movie Clip is 30 frames in length, and we place an Instance of circular on the Stage, and extend the key frame's range to 5 frames, followed by another range of frames for other content, the Movie Clip will not play through it's entire range of frames. It will only play through 5 frames. The Main Timeline has precedent over Movie Clip Timelines.



The circular Movie Clip is key framed in Frame 1 and extended to 5 frames. Frame 6 on the Main Timeline contains a key frame with the spiral Movie Clip, extended to a range of 90 frames. As the movie plays, only the first 5 frames of the circular Movie Clip are played.

As we cover Interactivity in Flash, you'll learn ways to control the Timeline so that the Marker can actually stop and sit at different points on the Timeline. Before that however, we need to examine the final type of Symbol, the Button.

Exercises

1. Create a Movie Clip of the letter A, capitalized. Make the letter rotate clockwise 360 degrees two (2) times.
2. Remember that you create the static Symbol first, a letter A as a Symbol with Graphic for Behavior.
3. Next you create a New Symbol with the Library, and animate the Instance of the letter. Remember to select Movie Clip for Behavior.
4. Place the Movie Clip on the Stage and preview it by pressing [CTRL] + [ENTER].
5. Animate the Movie Clip on the main Stage itself by moving it from one corner of the Stage to the other

Review Questions

1. What is the main advantage of building your animations in Movie Clips instead of on the Main Timeline?
2. Describe the process of creating a Movie Clip.
3. How do you know you are working on a Symbol Timeline rather than the Main Timeline?
4. To view Movie Clips placed on the Main Timeline, what must you do?
5. What purpose does a parent object serve in computer animation?

Summary

As a result of this chapter, you should be able to

- Create Movie Clips
- Use Multiple Timelines
- Understand Parent and Child Objects
- Use Movie Clips on the Stage

CHAPTER 9

Buttons



Buttons enable the user to interact with your Flash movie. With buttons, you can give users the power of choice. Users can navigate to different parts of your Flash movie, to different HTML files, and even control animations within Flash. Before you can add interactivity however, you have to build your buttons. Buttons are a unique type of Symbol. Like Graphic and Movie Clip Symbols, Button Symbols have their own Timeline and Stage, but the Timeline is very different.

OBJECTIVES

Upon completing this section, you should be able to:

1. Understand the Button Timeline
2. Understand Button States
3. Use Movie Clips in Buttons

The Button Timeline

Buttons are a unique type of Symbol. The Button Timeline differs from the regular Timeline in Flash in that each key frame contains a particular button state. In Flash, buttons, because they are activated by the mouse, can appear in three distinct states, Up, Over, and Down. The Button Timeline is divided into each state, with an additional state Hit, which defines the area of the button within which the user's mouse pointer becomes active.



The Button Timeline.

The Up State

The Up State is the normal state of the button as it appears when the user is not interacting with the button. Think of it as the way the button appears initially on the Stage.



The button as it appears in the Up state on the Stage.
When the user is not using the mouse on the button.

The Over State

The Over State is the appearance of the button when the mouse pointer passes over the button.



The button as it appears in the Over state on the Stage
when the user passes the mouse pointer over the button.

The Down State

The Down State is the appearance of the button when the mouse pointer is over the button and the user presses the mouse button down, or *{Clicks}* the button.



The button as it appears in the Down state on the Stage
when the user ***{Clicks}*** the mouse
button.

The Hit State

The Hit State is invisible to the user. The Hit state defines the area within which the mouse pointer becomes active. In this case, we would place a key frame of the ball in the Hit State to designate that the area of the ball is where the mouse pointer is active, and where, therefore, the user can use the button. By default, if you do not include a Hit State, Flash attempts to do its best to use the Instances you've placed in each State, however, to be definite, you should always define the Hit State

Creating Buttons

Buttons are Symbols made up of Instances of other Symbols, both Graphic and Movie Clips. To create a button, first you have to decide what the button will look like, and how it will behave in each State. You can place Graphic and Movie Clip Instances within each State to create static or animated buttons. Build your static and animated Symbols and then place Instances of those Symbols within the Button Symbol.

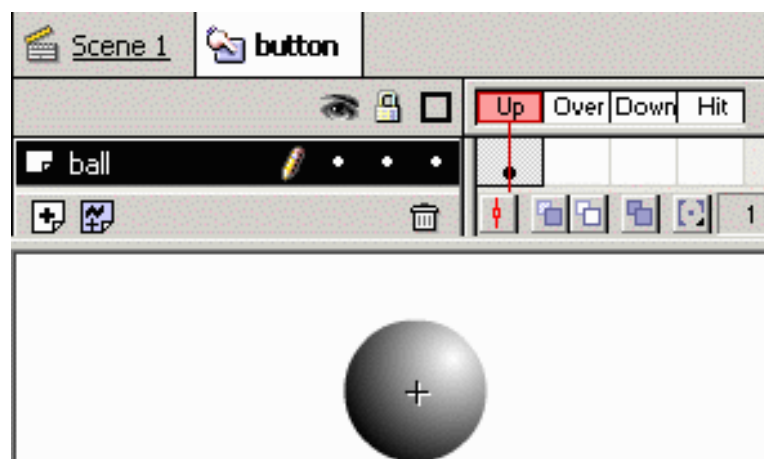
Lets take a look at what we want our button to look like.



To build our button, we begin by creating a New Symbol in the Library. {Click} the New Symbol icon . Name the Symbol appropriately and select Button for Behavior.

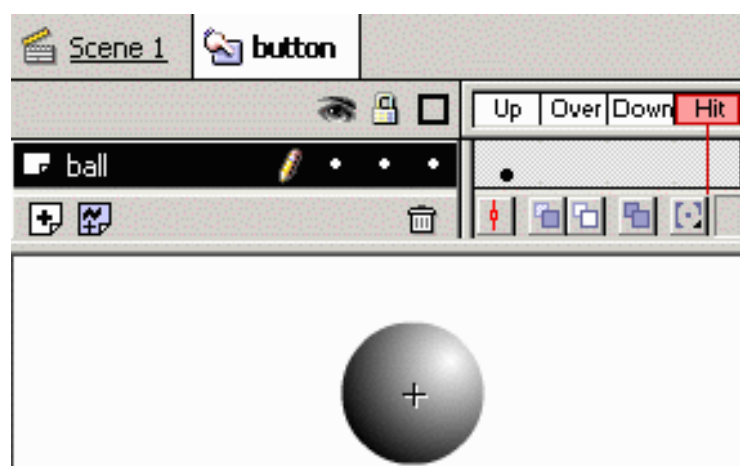


{Click} OK. You are placed into the new Symbol. We approach the creation of a button by thinking in terms of what Instances are going to be used, and therefore displayed, in each State. In this case, we have the ball Instance in each of our states. Just as you normally do, we begin with the initial state, in this case the Up State. It's a good idea to name our Layer, ball. By default, Flash places a blank key frame in the Up State. We simply drag the ball Instance on to the Stage and place it accordingly.



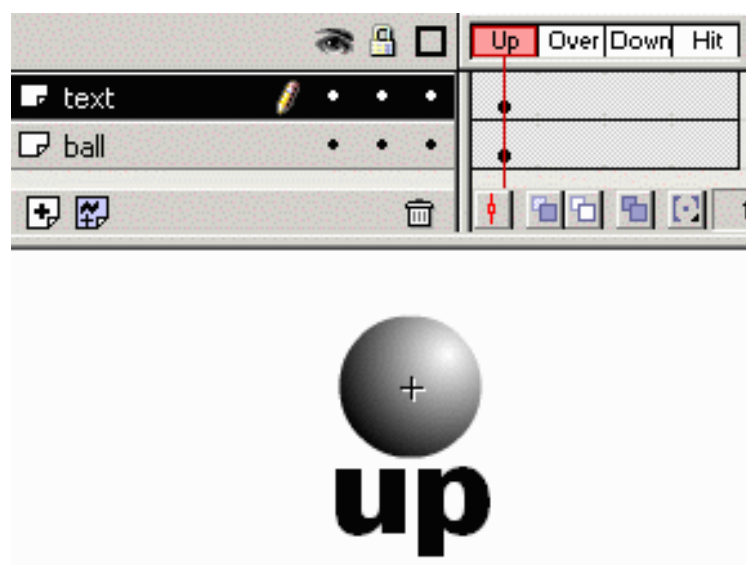
The ball Instance placed on the Stage in the Up State.

The ball appears in each of the remaining States (Over and Down). We have two options. We could insert key frames in each State. Each key frame would be a duplicate of the previous key frame. Or we could extend the range of the first key frame over the remaining States. Both methods are valid. We'll also be sure to key frame or extend into the Hit State. The Hit State is invisible to the user, whatever you place into the Hit State becomes the area where the mouse pointer becomes active.



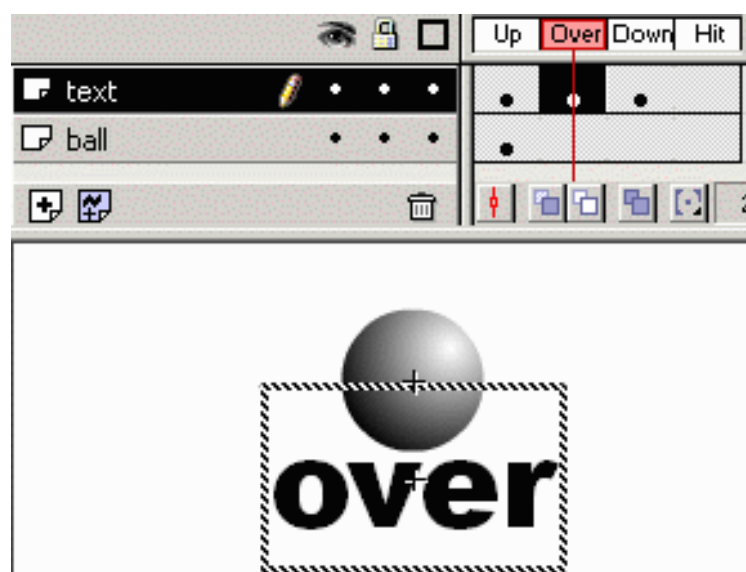
The range of the first key frame is extended across each State.

Our next step is to place the text Instances in each State. We create a New Layer, name it text and begin with the Up State. In the Up State we place the up Instance just below the ball Instance on the Stage.

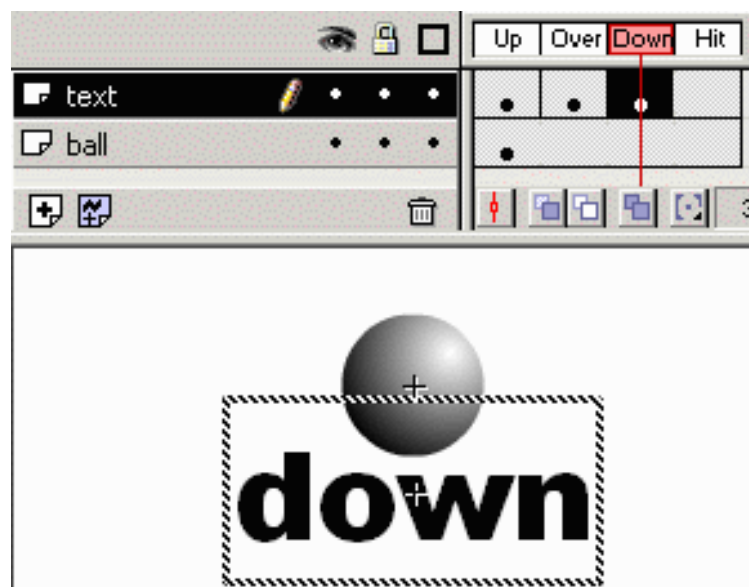


The new Layer text with the up Instance placed in the key frame in the Up State.

We can place multiple Instances on the same Layer because we're not Motion Tweening them. By default, when we create a new Layer in Flash, it automatically extends the range of the new Layer to match the previous Layer. As a result, our up Instance is extended across each State. We need to insert key frames in the Over and Down States, delete the up Instance from each, and place the over Instance in the Over State, and the down Instance in the Down State.



The text Layer will hold all of our text Instances in each State. Insert key frames in each State and place the appropriate Instance in each key frame.



The down Instance placed in the key frame of the Down State on the text Layer.

We don't need anything else but the ball Instance in the Hit State. Usually, you use the Instance which makes up the predominant shape of the button in the Hit State. In this case, we merely need to delete or remove the frame from the Hit State on the text Layer.



By examining the Timeline, we see that the ball Instance is extended across each State, and that the text Layer has key frames in each State but the Hit State. This indicates that we have separate Instances in each key frame on the text Layer, and that the ball Instance also defines the active area of the mouse pointer.

We never actually use the Button Timeline to build animations, all animation in buttons are Movie Clips placed within the key frames of particular States.

Using Buttons on the Stage

You use buttons just as you do any other Symbol, by creating a Layer, inserting a key frame in the Layer, and placing the Instance of the button on the Stage. Keep in mind that because buttons are Symbols, you can animate them, and apply any Instance properties to them as you need within your Flash movie. To preview buttons in the production window, select Control/Enable Buttons. Each of the States will display properly. If any of the States contain a Movie Clip however, then you must preview the Shockwave version of the Movie by pressing [CTRL] + [ENTER].

Control	Libraries	Window	Help
Play		Enter	
Rewind		Ctrl+Alt+R	
Step Forward		>	
Step Backward		<	
Test Movie		Ctrl+Enter	
Test Scene		Ctrl+Alt+Enter	
Loop Playback			
Play All Scenes			
Enable Frame Actions		Ctrl+Alt+A	
✓ Enable Buttons		Ctrl+Alt+B	
Mute Sounds		Ctrl+Alt+M	

Once we have created buttons and placed them on the Stage, we can add interactivity. Interactivity requires the use of Actions, which we cover in the next chapter.

Open the file *Button.fla* from the *Chapter09* folder and view the animation.

Exercises

1. Create a button.
2. First plan your button. Outline what you want in each of the States.
3. Create all of the Static elements that will be placed in the various Button States.
4. Create any Movie Clips that you will place in various Button States.
5. Finally, create the button.

Review Questions

1. What does the Hit State of a button do?
2. In a button, which State is displayed when the user passes the mouse over the button Instance?
3. How do you add animation to buttons?
4. Describe the process of planning and creating a button.

Summary

As a result of this chapter, you should be able to

- Understand the Button Timeline
- Understand Button States
- Use Movie Clips in Buttons

CHAPTER 10

Interactivity: Actions



Interactivity is a core feature of Flash. The elements necessary to build interactive movies are buttons and Actions. This chapter details how to put buttons to use, and how to use both Frame and Button Actions.

OBJECTIVES

Upon completing this section, you should be able to:

1. Understand and use Frame Actions
2. Understand and use Button Actions
3. Use Labels on the Timeline

Frame Actions and the Timeline

By default, when a Flash movie or Movie Clip plays, it loops. The marker or play head passes over each frame in the Timeline at the specified frame rate, displaying the Instances within each frame, until it reaches the end of the Timeline. At the end, it loops back to the beginning and proceeds again. This can be convenient, particularly for cyclical animations, such as the circular animation we created earlier, where a ball circles around a point, as if in orbit around an invisible planet. Because looping is a default, we need only have the Instance circle around once, and the cycle will repeat. However, there are times when we might not want an animation to loop, but rather stop when it has completed.

Preview the Actions1.swf file in the Chapter10 folder.

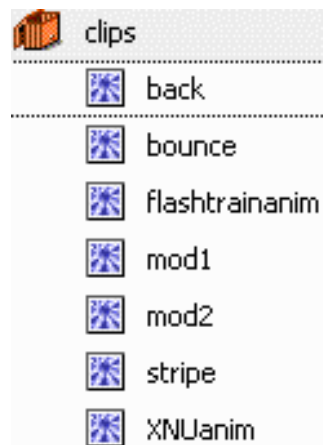
The movie consists of an animated logo (xnu.com), an animated Flash Training Movie Clip, animated black bars coming in from the sides of the screen at the top and bottom, and some buttons popping up. Because the Main Timeline and Movie Clip Timelines loop by default, this particular movie, in its current form, is chaotic. The key to controlling the Timeline is through the use of Frame Actions.

Using Frame Actions

Before we consider using Frame Actions on the Main Timeline, it's necessary to begin with Frame Actions in Movie Clips.

Open the Actions1.fla file in the Chapter10 folder.

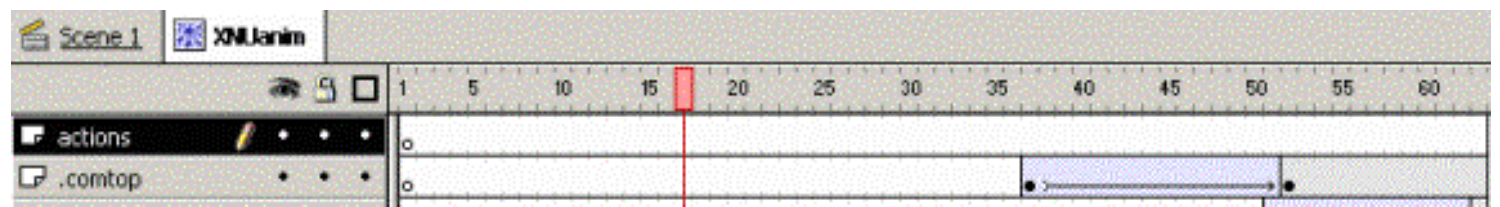
Open the Library and expand the clips folder. There are seven Movie Clips (back, bounce, flashtrainanim, mod1, mod2, stripe, XNUanim).



The Movie Clips in the clips folder.

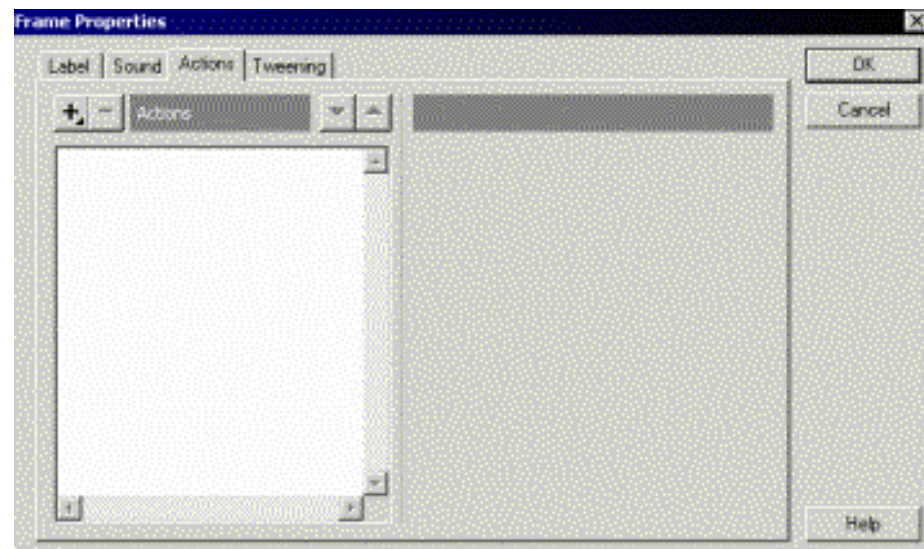
The first Movie Clip we're interested in is XNUanim. Edit the Movie Clip and play it (press [ENTER]). This is a logo animation. Each segment of the logo comes into view until the entire logo is complete. Typically, with logo animations, we are interested in the animation occurring, and then the completed logo remaining in view. When we place this Movie Clip on the Main Timeline, in a single frame or extended to match the number of frames in the Movie Clip, the Movie Clip will loop. How do we stop it from looping? We use a Frame Action called Stop.

Frame Actions are commands attached to key frames on the Main Timeline. Before doing so, a popular technique that many Flash developers use is creating a Layer dedicated to holding only Frame Actions. In this example, we are going to add a Layer to the very top of the XNUanim Timeline and call it actions.



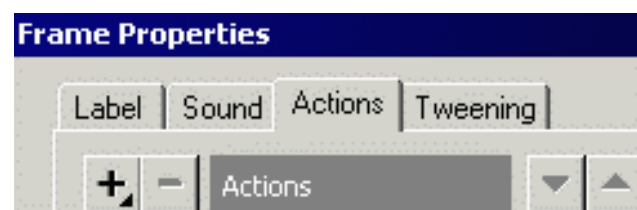
A new Layer is added just above the .comtop Layer. We label it actions.

This particular animation is 63 frames in length. We want to stop the Marker from looping back to the beginning and playing the animation again. We need to add a Frame Action. First, we insert a key frame on the actions Layer at Frame 63. This key frame will not store Instance information, we use it for Frame Actions. To attach an Action to a key frame, *{Double-Click}* the key frame. The Frame Properties dialog-box pops up. *{Click}* the Actions tab.

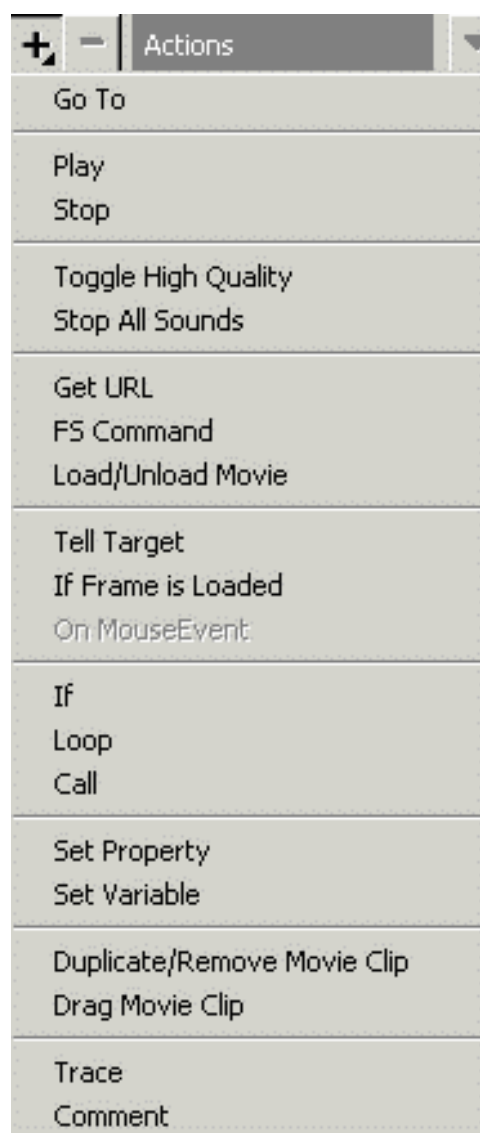


Selecting Actions

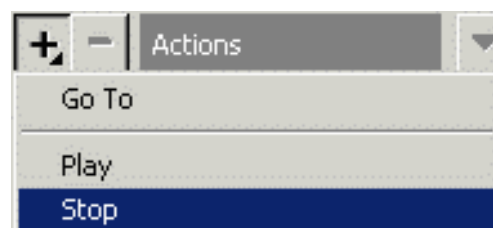
To select or add Actions to a Frame, *{Click}* the add icon .



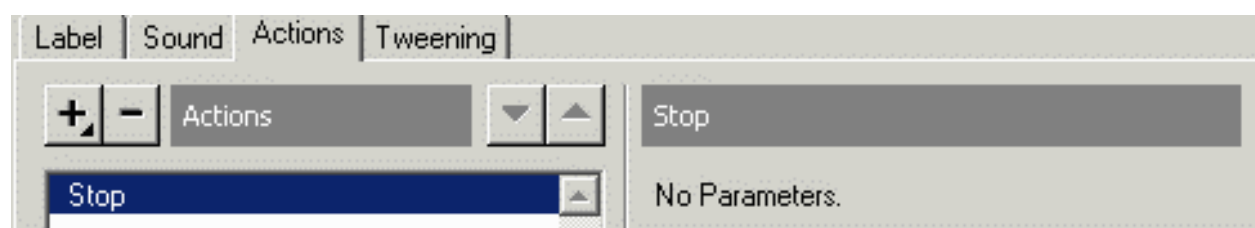
A list of possible Actions appears.



To select an Action, simply *{Click}* on it in the list, much like selecting an item from a standard menu. In this case we select Stop.



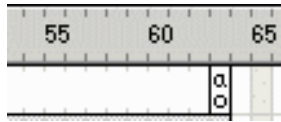
When you select an Action, the Action is identified, and a list of possible parameters appears on the right side of the dialog-box. The Stop Action does not have any parameters. Actions we use later in the chapter do have parameters.



The left side of the dialog-box lists the Action or Actions we've selected. The right side of the dialog-box lists possible parameters for the selected Action.

{Click} OK to add the Stop Action to the key frame. In the Timeline, a small

letter a appears above the key frame indicating that a Frame Action is attached.

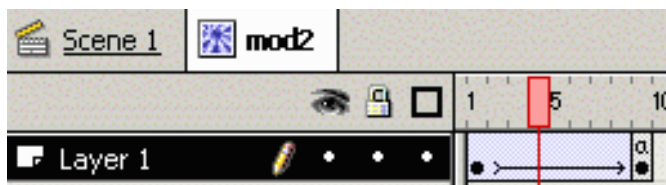


A small a appears above the blank key frame in Frame 63 of the actions Layer.

Open the Actions2.fla file in the Chapter10 folder.

Preview the movie by pressing [CTRL] + [ENTER], and watch the XNUanim Movie Clip. Notice that it stops as it completes and remains in view, even when it moves up and to the left later in the movie. Previously, it looped the moment it was complete and continued to do so throughout the movie.

Our next step is to edit each Movie Clip in turn and add a Stop Action at the end of each animation. Note that you do not have to add a actions Layer. For instance, the mod2 Movie Clip consists of one Layer, Motion Tweened across nine (9) frames. You could simply {Double-Click} the last key frame in the Layer and add a Stop Action to it rather than adding a new Layer, inserting a key frame at the end of the Layer, and adding a Stop Action to it.



A Stop Action is added to the last key frame.

Both methods are valid. Some developers find creating a separate actions Layer helps keep them organized and productive. Others use them in complex animations, and don't bother to use them for simple animations such as the mod2 Movie Clip.

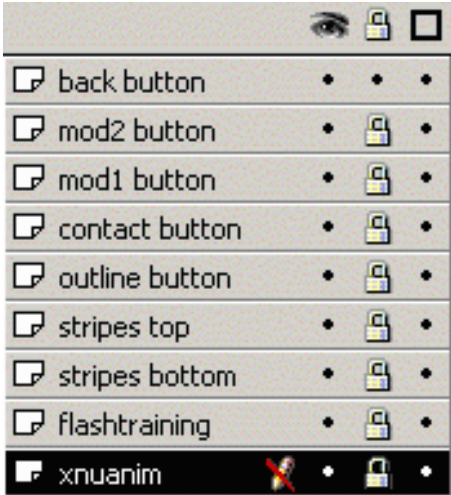
Open the Actions3.fla file in the Chapter10 folder.

Once you have added a Stop Action to each of the Movie Clips, preview the movie again (recall that the preview mode isn't the most accurate view, you may see orange above and below the black stripes in the animation). If you watch carefully, you'll notice that all the Movie Clips play through their animation sequences and then stop. However, the Main Timeline is still looping. We need to add Stop Actions to the Main Timeline.

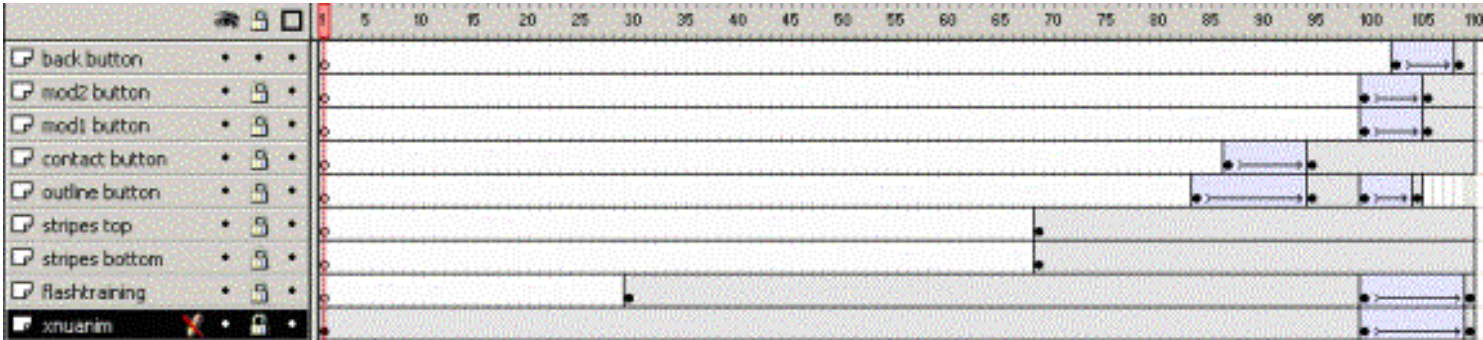
Before we add Actions to the Main Timeline, it will be useful to outline what the objective of this particular movie is. The logo animates and rests in the center of the Stage. The words Flash Training appear animated below the logo and also come to a stop. Black stripes move above and below the logo, coming to a rest. Two buttons, outline and contact appear in the black area below and to the right of the logo. The user is supposed to view the animations and then {Click} on either outline or contact as they wish. Presently, the movie does not stop when the animations are complete and both buttons appear. It continues on and into the outline section of the movie. Let's examine the Timeline.

First, we'll look at the Layers. Each Layer contains an Instance.

At this point, we have nine (9) Layers in total.



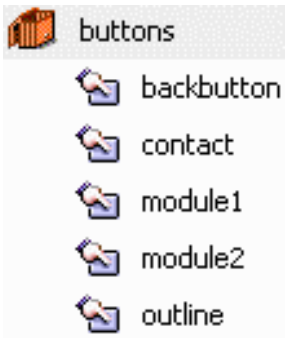
If we examine the Timeline, we see that majority of the animation is done by the Movie Clips.



The stripe Movie Clip occupies both the stripes bottom and stripes top Layers. There is only one key frame on each of these Layers extended forward. The Movie Clip performs the animation.

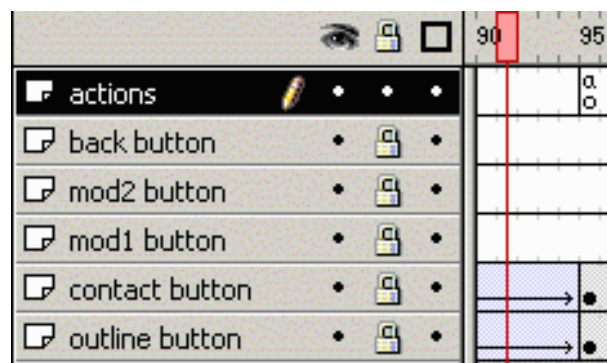
The XNUanim Movie Clip occupies the xnueanim Layer. There are three key frames. The first key frame at Frame 1 plays through the animation and because we've put a Stop Action inside the Movie Clip, it stops. The second key frame and the third key frame combine in a Motion Tween that repositions the Movie Clip and reduces it in scale. Because we have stopped the Movie Clip on the last frame of the animation, this is what the viewer sees from this point on. Similarly, the flashtrainanim Movie Clip on the flashtraining Layer stops and is rescaled and repositioned later in time. Recall that in order for both these Movie Clips to fully animate through to the end, we've extended the range of their first key frames to at least match, if not surpass, the amount of frames in each Movie Clip's Timeline.

The outline button and contact button Layer contain the outline and contact Button Instances respectively. To view a full list of the buttons in the movie, expand the buttons folder in the Library. There are five buttons.



The Button Symbols in the Library.

Because only Button Instances can contain Button Actions (which we will get to in a moment), we've animated the buttons on the Main Timeline. Hence both these buttons are animated, beginning at Frame 84 and 88, and both ending at Frame 95. It is at this point that we want the user to select where they'd like to go to next, either to the outline section of the movie or to the contact section of the movie. Because the Marker moves through each frame and loops automatically, the movie doesn't stop at this point. We need to place a Frame Action at Frame 95. To so, we'll make an actions Layer placed at the top of the Timeline, insert a key frame at Frame 95, and place a Stop Action on the key frame.



An actions Layer is inserted at Frame 95, and a Stop Action is applied to the key frame.

Open the Actions4.fla file in the Chapter10 folder.

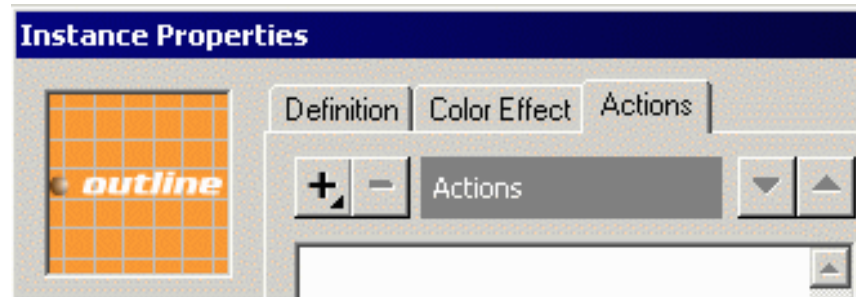
Now when you preview the Shockwave Movie, the Movie Clips play through and stop, and the entire Movie plays and stops at a point where both the outline and contact buttons are ready to be {Clicked} on.



The movie pauses at Frame 95 on the Main Timeline.

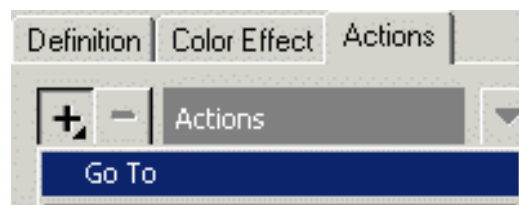
Button Actions

In order to jump to the outline section of the Movie, which begins at Frame 100 on the Timeline, we need to jump from Frame 95, where we've stopped, to Frame 100, where the animation that presents the Outline information occurs. In order to jump from one point to another, we need to attach a Button Action to the outline instance at Frame 95. To attach a Button Action to a Button Instance, *{Double-Click}* the Instance on the Stage and select the Actions tab from the Instance Properties dialog-box. Do not confuse Button Actions with Frame Actions. Frame Actions are accessed by ***{Double-Clicking}*** on a key frame. Button Actions are accessed by ***{Double-Clicking}*** on the Button Instance on the Stage.



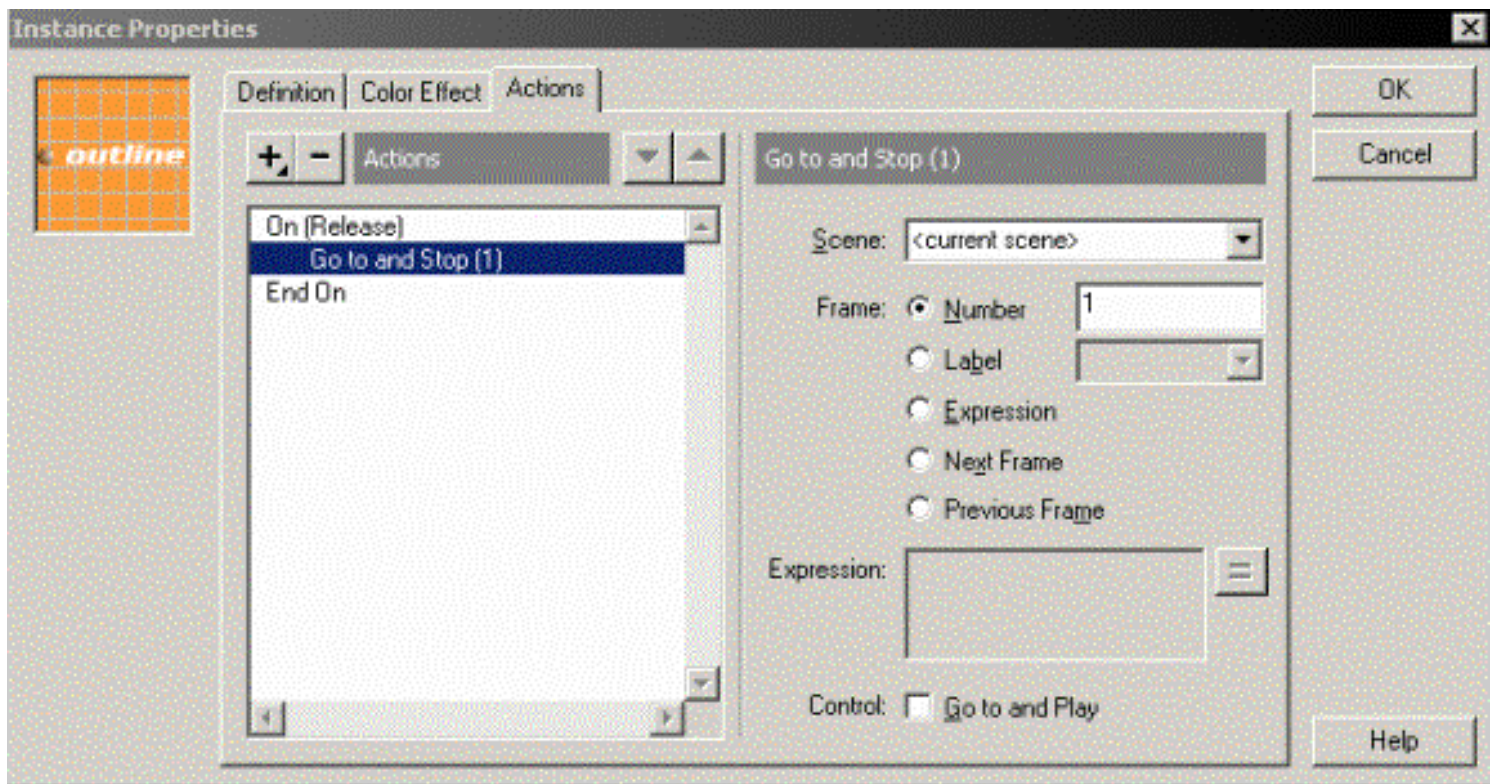
Actions are available to Instances with Button as the Behavior.

When we need to go to another location on the Timeline we use the Go To Action. Select the Go To Action from the Actions menu.



Use the Go To Action to jump to other locations on the Timeline.

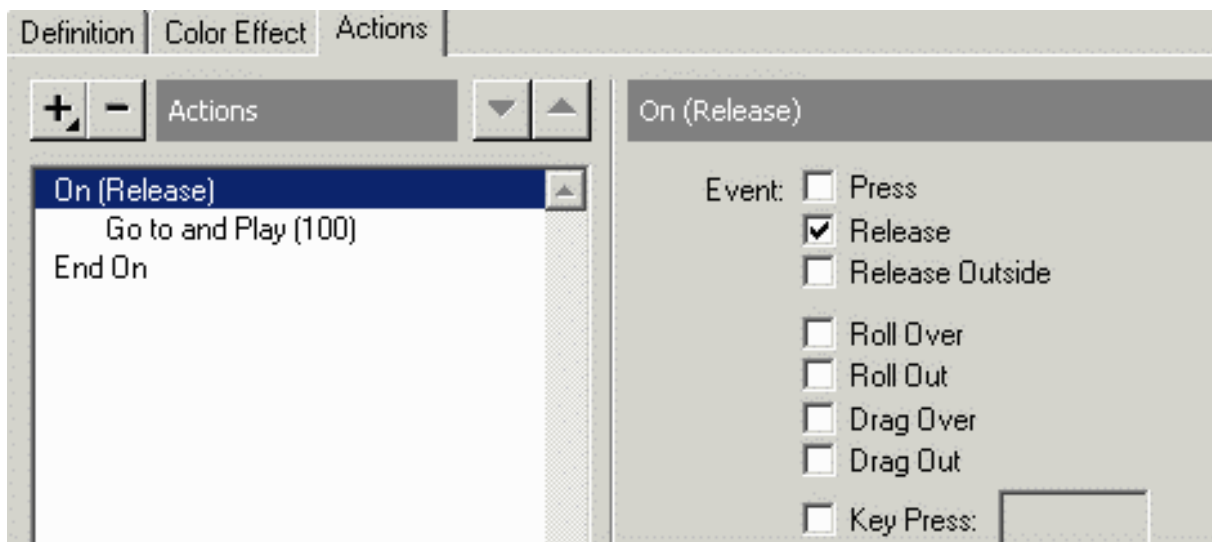
When you select the Go To Action the following parameters are available:



Selecting the Go To Action presents you with a number of parameters.
The On Event

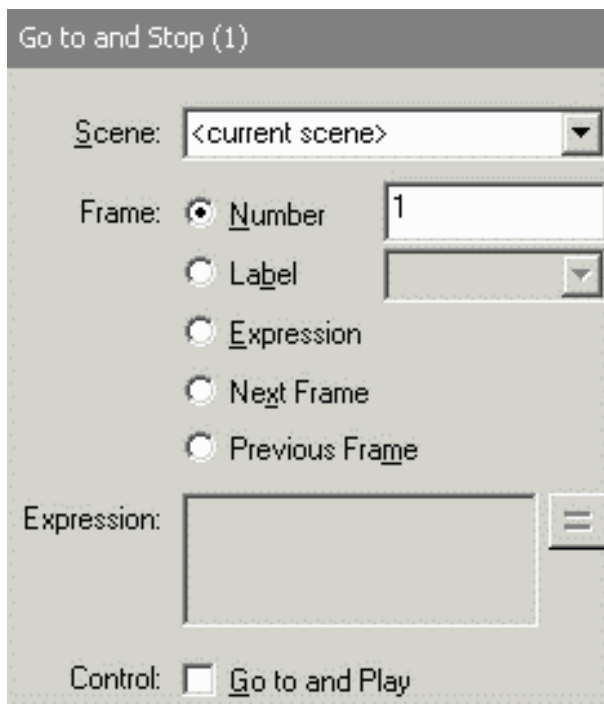
The Go To Action is surrounded by an On event, in this case On (Release) and End On. An On Event isolates a particular mouse action, in this case Release, which means when the user releases the mouse button after {Clicking}, whatever is listed within the On Event (between the initial On and the End On will occur.

To change the On event, simply {Click} the On (Release) line of code in the Actions list and select a different mouse event from the listed parameters.



Flash presents you with seven Mouse Events and a Keyboard alternative.

By default, when using a Go To Action, Flash places a Go to and Stop (1) command. If you examine the Parameters for the Go To Action, they are divided into Scene, Frame, Expression, and Control.



Scene

If your Flash Movie has multiple Scenes, you can select which Scene to jump to with the drop-down menu. By default, the current scene is selected. Our movie has only one scene.

Frame

Use the Frame setting to specify the Frame you wish to jump to. You can specify Frame number by checking the Number radio-button and inputting a Frame Number into the Number field. You can specify a Frame Label if you have labeled a Frame by checking the Label radio-button and selecting the Label from the drop-down menu or inputting the Label into the Label field. We cover Frame Labels later in the chapter.

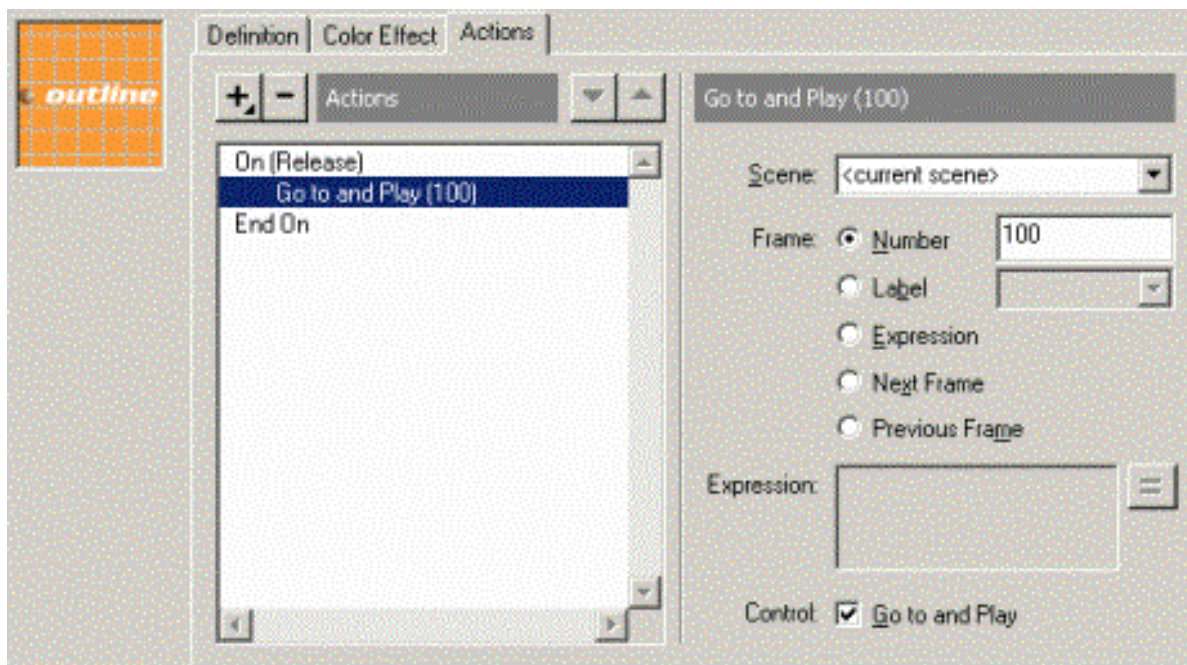
Expression

Check the Expression radio-button and input an Expression into the Expression field. Expressions are part of Flash programming, covered in the Flash programming supplemental volume.

Control

Check the Go to and Play checkbox to force Flash to jump to the Frame you specify and cause the Marker to play from that point on. If you do not check Go to and Play, the Marker will jump to the Frame you specify and stop at the Frame, as if a Stop Action has been placed at the Frame.

We need to go to Frame 100 in our Movie. We check Number in Frame and input 100. We also check Go to and Play in Control to ensure the Marker plays from Frame 100 onward.



A Go To command to Frame 100 with Go to and Play checked. Note the syntax, the Frame number is placed in brackets Go to and Play (100).

Open the *ButtonActions1.fla* file in the *Chapter10* folder.

Having made the changes, when we preview the Shockwave movie (by pressing [CTRL] + [ENTER]), the movie plays and stops, waiting for user interaction. {Clicking} the outline button causes the movie to proceed. The logo reduces in scale and moves to the left, the outline button fades away, and a module1, module2, and a back button appear. However, the movie still loops and goes back to the beginning. We need to place a Stop Action at the end of the outline section. To do so, we place a key frame at Frame 110 of the actions Layer and place a Stop Action on the key frame.

Open the *ButtonActions2.fla* file in the *Chapter10* folder.

Now when we play the movie, after {Clicking} the outline button, the movie jumps to the outline section and stops again, waiting for the user to {Click} either the module 1 or module 2 button, or the back button to return to the main section of the movie. At the moment, none of these buttons have Actions attached to them. We're going to make the back button active by attaching a Go To Action to it. Instead of using Frame Numbers, we're going to use Frame Labels.

Frame Labels

Using Frame Numbers can quickly become confusing and unreliable. Any change in the beginning point or end point of a particular animation, or a group of animated Layers, can change the Frame Number(s) referenced in Actions. The best method of referring to Frames is to label them. To label a Frame, {Double-Click} a key frame and select the Label tab from the Frame Properties dialog-box. In this case, we've added a key frame to the actions Layer at Frame 100, and input outline as the Label.



Applying a Label to a key frame.

By default, Label is checked in Behavior. Labels are addressable by Actions, such as the Go To Action. If you check Comment, the text you input becomes a comment to you, the developer, and cannot be used or addressed by Actions. Both Labels and Comments appear in the Timeline. Labels are distinguished with a flag above the key frame. Comments are distinguished with two dashed lines above the key frame.

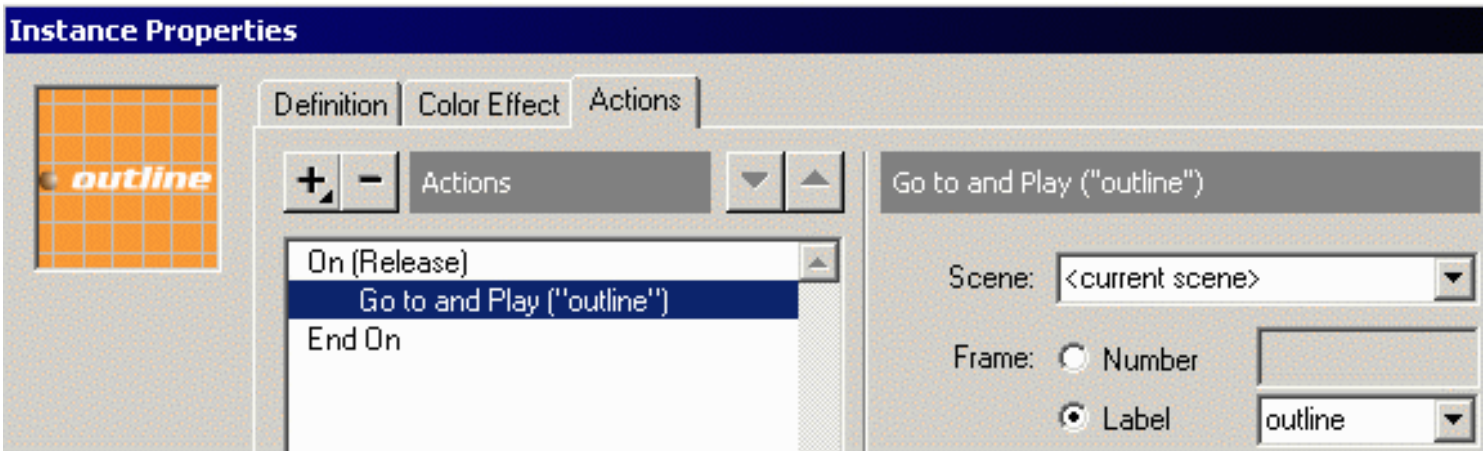


The key frame at Frame 100 is labeled outline.



The key frame at Frame 100 is commented outline.

Now we've clearly labeled the beginning of our outline section. It's a good idea to label each section of your movie on the Timeline. We need to label the very first key frame of the actions Layer start in case we want to jump to the beginning of the movie. As well, we should label the point at which we want the back button to jump back to the main menu, in this case at Frame 95, where the first animation sequence stops. We'll label this key frame main. Furthermore, we should return to our outline button and change the Go To Action from Frame (100) to Frame (outline). To do so, simply {Double-Click} the outline button on the Stage, select the Actions Tab, check Label and select outline from the drop-down list.



The outline button is changed from referencing Frame 100 with the Go To Action, to the Frame Label outline, which, at this point, is placed at Frame 100. We could move the whole outline section, and all the animations on each Layer, to different Frames on the Timeline and the button will still go to the Label, rather than a specific Frame.

If we add a Go To Action to the back button within the outline section, and target the Go To to

the Label main, when we {Click} the back button, the movie returns to the point at which the Logo appears in the middle of the Stage, and both our outline and contact buttons are available as options.

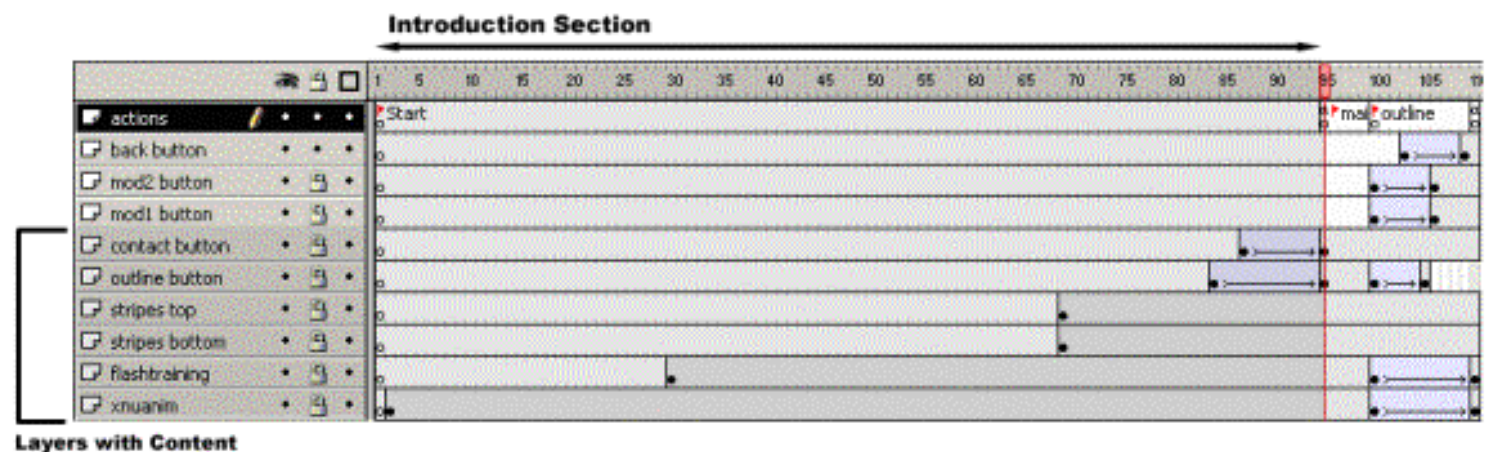
Open the Labels.fla file in the Chapter10 folder.

Movie Structure

Let's examine the structure of this movie. Before creating the movie, we would have storyboarded the structure of the web site and the animation. We have three main sections: Introduction, Outline, and Contact.

Introduction Section

The Introduction consists of the Logo animation, the Flash Training animation, the Stripes animation, and the animated buttons, outline and contact.



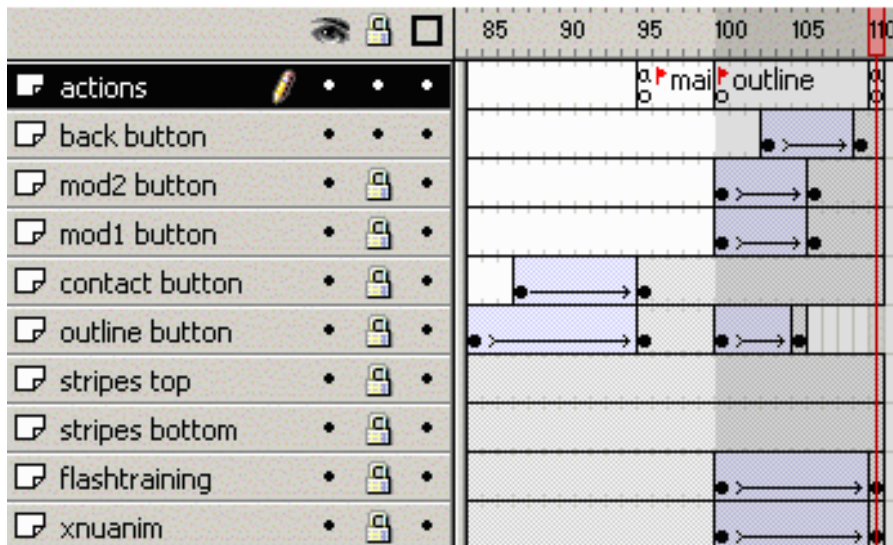
The Introduction Section on the Timeline.

On the Timeline, the Introduction Section begins at Frame 1 and ends at Frame 95. In the actions Layer, Frame 1 is labeled Start. At Frame 95, we have a Stop Action, and the key frame is labeled main, which will act as a point to jump back to and from the other sections in the movie. Note that the Layers that do not have content within this section, have content within the next section, outline, starting at Frame 100. By using Layers, Frame Actions, Labels, and Button Actions, we can divide the Timeline into particular sections. Frame Actions, Button Actions, and Labels, combined with Button Instances, permit us to jump back and forth between sections. We place content within each section on Layers, at particular points in time on the Timeline.

Outline Section

The Outline Section consists of the module 1, module 2, and back buttons for new content. The contact button is extended from the previous section into this section. The outline button is animated in this section, and fades from view (Frames 100 to 105). Finally, the content on the flashtraining and xnuanim Layers moves to the upper left of the screen and reduces in scale. A Stop Action has been placed at the key frame of Frame 110 in the actions Layer to stop the Marker as the animation within this section is completed, permitting the user to {Click} on the buttons within this section.

Outline Section



The Outline section on the Timeline.

We will build the Contact section in the exercises for this chapter. As you can see, Flash movies, and individual sections of Flash movies are presented across the Timeline, with content and animations occurring at various points in time. Planning is essential.

Exercises

We need to link to the contact section to our movie.

1. Open the file Exercise.fla from the Chapter10 folder.
2. We've add a contact section the begins at Frame 115.
3. Add the necessary labels, Frame Actions, and Button Actions to make the necessary jump from the end of the start section (which ends at Frame 95, labeled main) to the area of the Timeline that displays the contact information.
4. Examine the file Contact.fla. This is the completed file for the exercise.

Review Questions

1. By default, what happens when the marker comes to the end of the Timeline?
2. Explain the procedure to prevent Flash from looping?
3. When using Actions, Actions appear on the left side of the dialog-box. What appears on the right side?
4. What type of Instance can Actions be attached too?
5. What is the default On event when using the Go To Action with a button?
6. How do you select a different On event?
7. What is the main advantage of using Labels?
8. How are Flash movies typically structured?

Summary

As a result of this chapter, you should be able to:

- Understand and use Frame Actions
- Understand and use Button Actions
- Use Labels on the Timeline

CHAPTER 11

Interactivity: Tell Targets



The Tell Target command is a powerful interactive element that permits you to control Movie Clip Timelines. With careful planning, Tell Target can be used to simplify the structure of the Main Timeline.

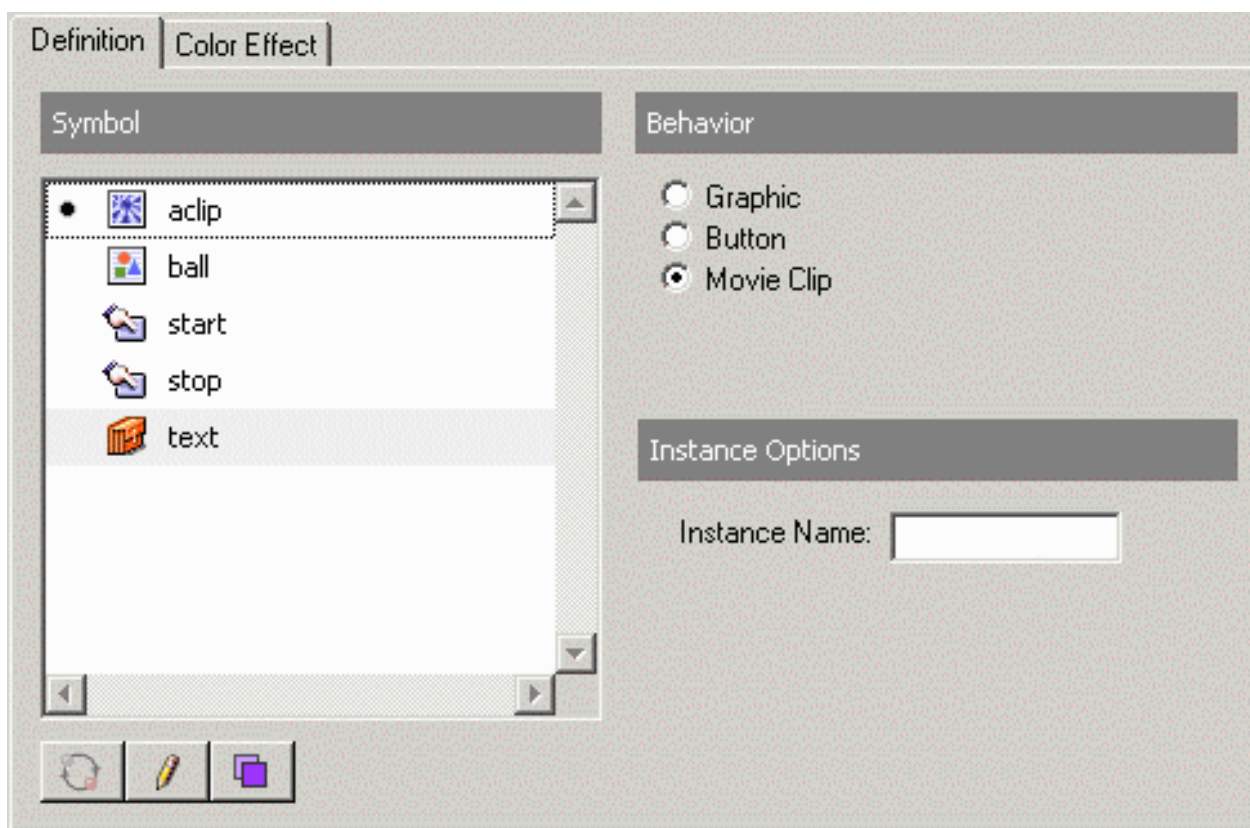
OBJECTIVES

Upon completing this section, you should be able to:

1. Designate a Movie Clip with a Unique Identification
2. Use the Tell Target Command to control Movie Clips
3. Understand the hierarchical structure of Multiple Timelines
4. Use the Tell Target Command to structure Movies

Unique Identities for Movie Clips

The Tell Target command targets specific Timelines. In order to target specific Timelines we need to identify them. In order to identify Timelines, the Instance must be a Movie Clip. To attach an ID or name to a Movie Clip Instance, drag the Instance on to the Stage and *{Double-Click}* it. The Instance Properties dialog-box pops up. *{Click}* the Definition Tab.



The Definition Tab. The selected Symbol is listed in the Symbol box. In this case aclip. Input an ID or name into the Instance Name field.

Name the Instance for Tell Targeting in the Instance Name field. The Instance Name can differ from the Symbol name.

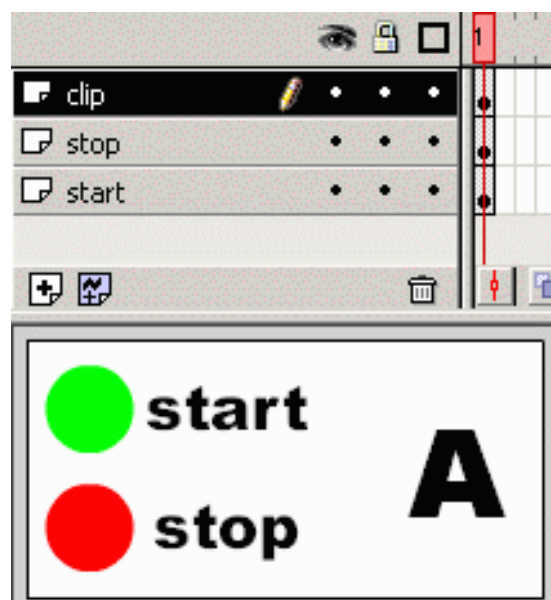
Instance Name:

In this case, we've named our Movie Clip Instance rotation. The name of the Symbol is aclip. Each Instance of the Symbol could have a unique ID or name. In order to control a Movie Clip Instance, you must give it a name.

Using Tell Target to Control Movie Clips

Once you have named a Movie Clip Instance, you can address it with the Tell Target Action. For example, we have a movie with a start and stop button that will start and stop the aclip Instance, which is an animation of the letter A rotating 360 degrees. The Instance has been named rotation.

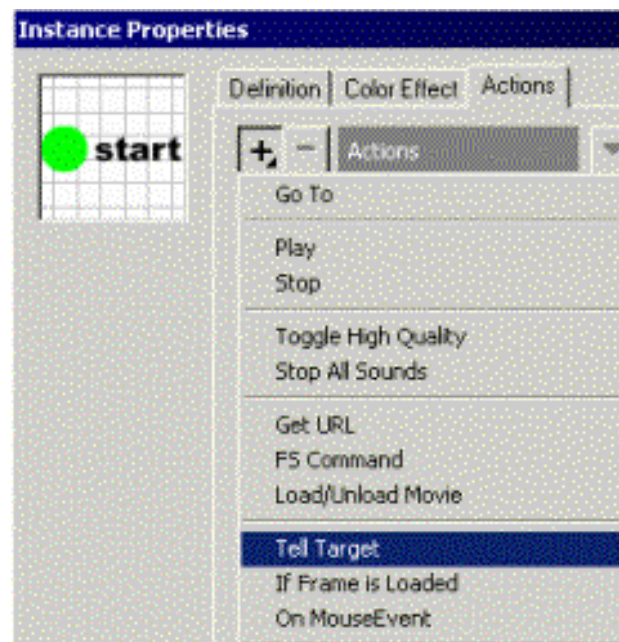
Open the file Target1.fla from the Chapter11 folder and follow the next steps.



The movie consists of start and stop buttons to start and stop the Movie Clip Instance.

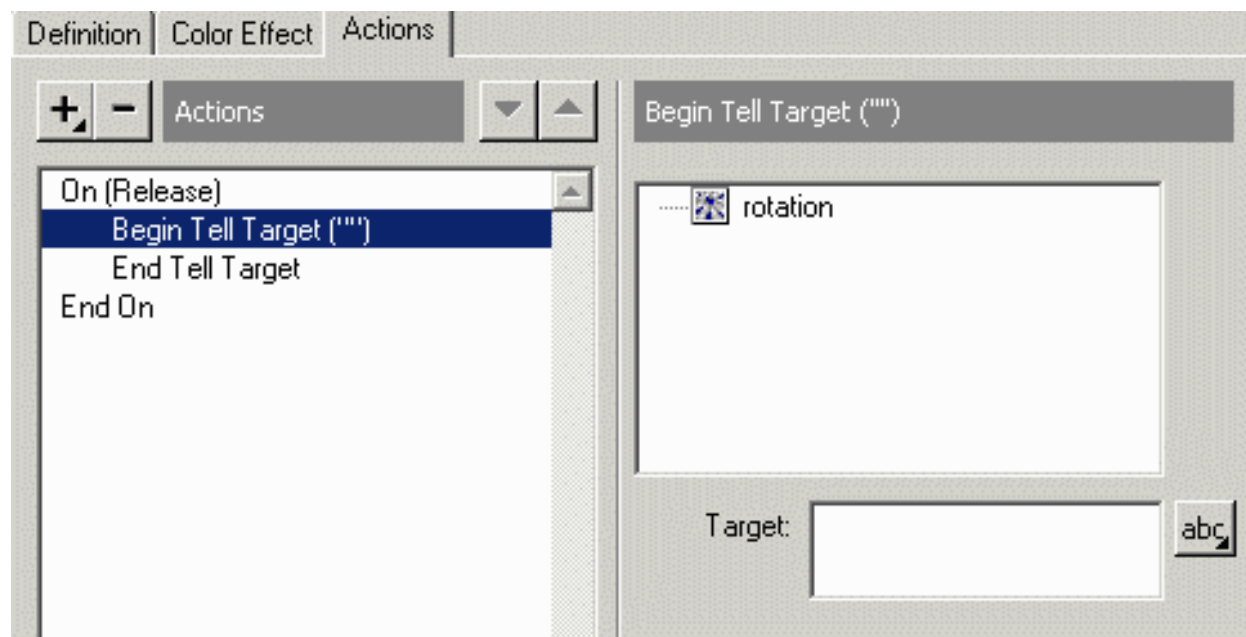
To target the Instance with Tell Target, we need to attach a Tell Target Action to each of the buttons, much like attaching a Go To Action. We'll start with the start button.

We *{Double-Click}* the button Instance on the Stage and *{Click}* the Actions Tab. From the available Actions, we select Tell Target.



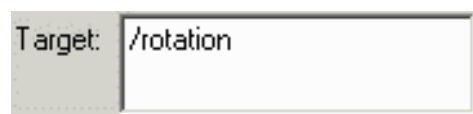
Select Tell Target from the Actions menu.

By default, Flash places an On (Release) event around the Tell Target command. Next you need to select or target the Movie Clip you wish to address or control. Think of it as targeting a Timeline. In this case we're going to target the rotation Timeline. A list of named Instances appears in the Begin Tell Target box.



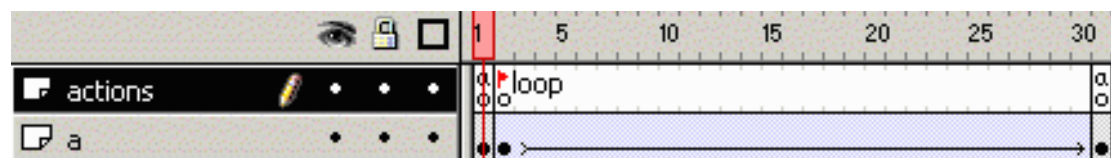
You must specify the Timeline you are targeting. Named Instances appear in the Begin Tell Target box.

{Double-Click} the name of the Instance you wish to target. The Instance name appears in the Target box.



You might notice the forward slash / in front of the Instance name. There is a hierarchy of Timelines when using Tell Target that we cover later in the chapter.

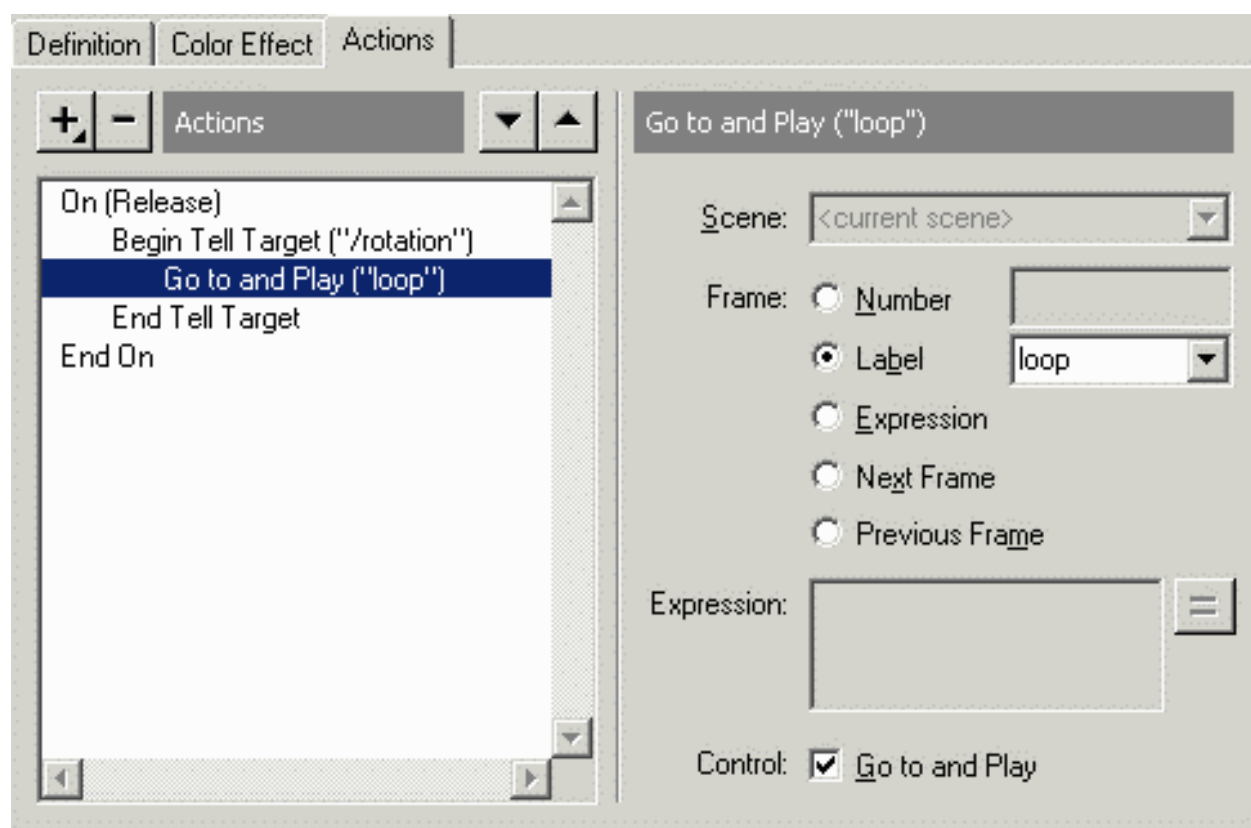
Once you have targeted the Timeline of the Instance you can add more Actions that affect the Timeline. In this case we want the animation to start playing. To do so, we use the Go To Action. When you use Tell Target, you must prepare the Timelines you wish to target appropriately. Let's examine the Timeline of the aclip Symbol.



The Timeline of the aclip Symbol.

The Timeline consists of the animation on the a Layer, and an actions Layer for our Actions. The first, second, and third key frames on the a Layer are identical, with the Instance of the a Symbol centered on Stage. The Motion Tween between key frame 2 and key frame 3 involves an automatic, clockwise rotation of 2 times. In the actions Layer, the first key frame has a Stop Action on it so that the Movie Clip doesn't automatically move to the second key frame and start looping on the Stage. The second key frame is labeled loop. This is where we want to go to when we target the Instance with our start button. The third key frame has a Go To Action which goes to loop and plays the animation again. We needed to place a Go To Action at this point to make the movie loop, or else the marker would return to the key frame at Frame 1, which has a Stop Action on it.

In our button, we select the Go To Action and check label. We need to input the label manually in the Label field. The Frame Label will not be present in the drop-down menu because the Label exists on a different Timeline than the Main Timeline, upon which our button is currently placed.

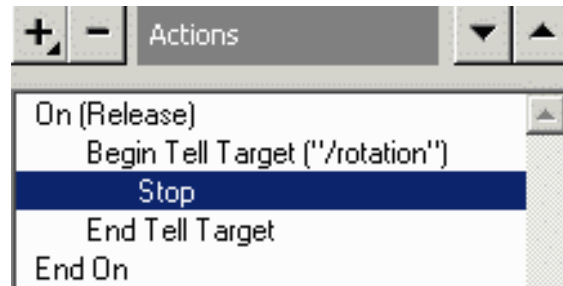


We've targeted the rotation Timeline (the Instance of aclip) with a Tell Target command. Having targeted the Timeline, we go to the key frame labeled loop, making sure to check Go to and Play in Control.

Now when we preview the movie (with [CTRL] + [ENTER]) and press the start button, the Movie Clip loops on the Stage.

Open the Target2.fla file in the Chapter11 folder to test the Tell Target if you haven't done it yourself..

To stop the Movie Clip from playing we can take two approaches. First, we need to target the Movie Clip with the Tell Target command. Having done that we can simply select Stop as our next Action. This will stop the animation immediately at the frame it is currently within. Alternately, we could Go To Frame 1 to have the Movie Clip return to the first frame, where a Stop action exists. The letter A will return to its original orientation.



Attaching this series of Actions to the stop button will stop the Instance instantly, at whatever point the animation is in the Timeline.



The A stops at whatever point its on in the aclip Timeline.

Open the Target3.fla file in the Chapter11 folder and examine the targetting effect of the stop button if you haven't added the Tell Target yourself.

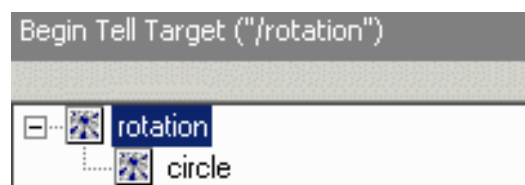
The Hierarchical Structure of Multiple Timelines

Timelines follow a hierarchical structure, precisely like the structure of folders on a hard drive. At the top of the hierarchy is the Main Timeline. Think of it as the main folder that contains a series of subfolders. Subfolders are Movie Clips, and Movie Clips within Movie Clips are like subfolders within subfolders. When using the Tell Target Action, its important to keep the hierarchy in mind. Recall that when we targeted the Instance we named rotation, the path to the target was included in the Target field.



The path to rotation.

The path structure of multiple Timelines follows the exact same syntax as navigating through folders to documents in HTML. For Instance, the rotation Timeline is one level below the Main Timeline, hence the path /rotation. If there were another Movie Clip embedded within the aclip Timeline, named circle for example, and you wished to address this Instance with a button, the path would be /rotation/circle. In the Begin Tell Target field, the named Instance appears below rotation because it is embedded within it, like a sub directory within a directory.



circle is one level below rotation.

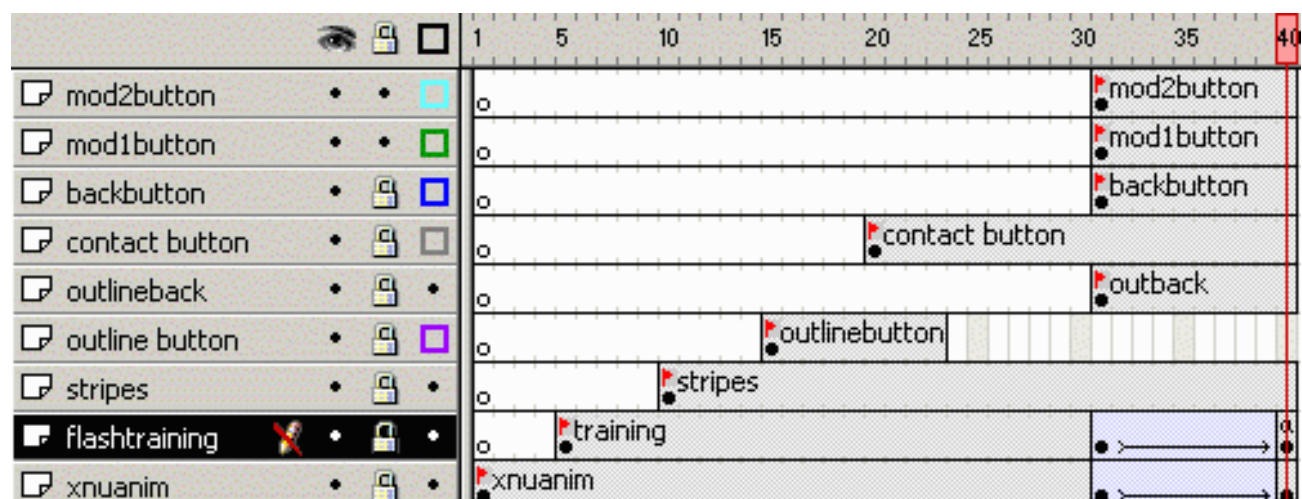
Likewise, if a button were embedded in the circle Timeline and you wished to address the rotation Timeline, the path would be ../ because we are stepping up one level, just as we do in HTML to step up from one folder to a folder a level up.

At all times, the path to the Main Timeline is simply / regardless of how many embedded levels you are addressing it from.

Using Tell Target to Structure Movies

With planning, you can use Tell Target to structure your movies, making for a clean main Timeline. In our Actions examples, the Main Timeline totaled 140 Frames. Using Tell Target, we can present the same animation in 40 Frames.

Open Structure1.fla from the Chapter11 folder. Preview the movie and exam the main Timeline.



The main Timeline of structure1.fla is composed entirely of Movie Clips.

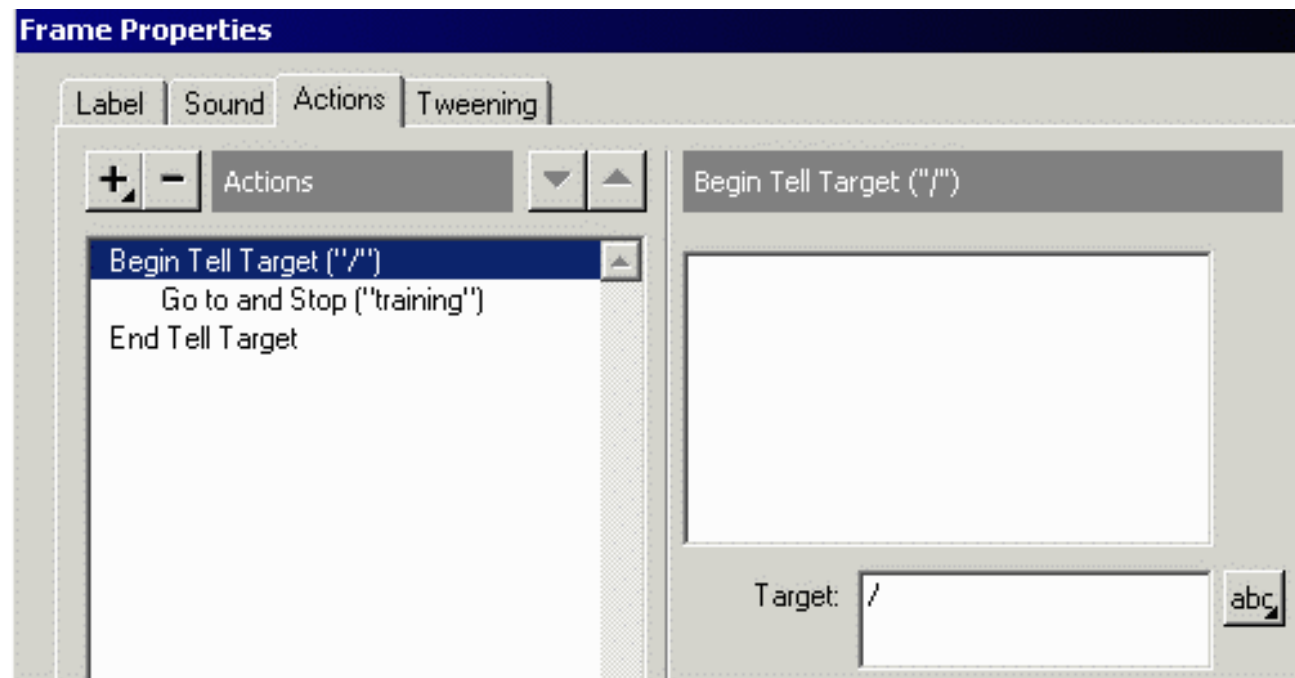
The first thing you'll notice is that the timing of the animation is off, with each Movie Clip running into the other. We can use a combination of Frame Actions, Frame Labels, and the Tell Target Action to control the timing of each animation, as well as use Button Actions to jump to different points in the Timeline.

First we need to place Stop Actions on each of the main Movie Clip animations so that they can run their course before the next clip begins. We apply Stop Actions to the key frames labeled xnueanim, training, stripes, outlinebutton, and contact button. In each case, each key frame contains a Movie Clip.

Open the Structure2.fla file in the Chapter11 folder if you haven't added the Stop Actions yourself.

Now, when we preview the movie, we see only the xnuanim clip play because we've placed a Stop on that frame. We can place a Tell Target in the XNUanim Symbol's Timeline, to activate the next point in the main Timeline. To do so, we edit the XNUanim Symbol.

At Frame 35 on the Actions Layer, we place a Tell Target on a blank key frame and send a Go To and Stop command to the label training on the main Timeline. The target address or name of the Main Timeline is always /.



Within the XNUanim Symbol, we place a Tell Target Action to the main Timeline that triggers the next Movie Clip.

Now when we preview the movie, the logo animation occurs, and as it continues, the next animation, Flash Training proceeds. Then the movie stops again because we've got a Stop Action on that key frame, and we've also sent a Go to and Stop Action from the XNUanim Symbol. To proceed, we'll have to place a Tell Target within the flashtrainanim Symbol to activate the next point on the main Timeline, in this case stripes. Because we are using Movie Clips, the clips themselves continue to play through, even though we are jumping or activating other clips at different points in the main Timeline. The Movie Clip will follow its own Timeline.

At Frame 37 in the flashtrainanim, we place a Tell Target Action to activate the clip placed at the stripes label on the main Timeline.

```
Begin Tell Target ("/")
Go to and Stop ("stripes")
End Tell Target
```

The label stripes is targeted on the main Timeline.

Finally, to complete the introductory animation to this particular Flash movie, we need to place a Tell Target Action in the stripe Symbol to activate the outline Movie Clip at the label outlinebutton on the Main Timeline as well as placing an additional Tell Target Action to activate the contact Movie Clip at the label contact button.



There are three Actions in the Actions Layer of the stripes Symbol.

At Frame 23, outlinebutton is targeted on the main Timeline.

At Frame 25, contact button is targeted on the main Timeline. These Tell Target Actions will activate the outline and contact Movie Clips. At Frame 26, there is a Stop Action to prevent the stripes animation from looping.

Within the outline Movie Clip is an Instance of the Button outline. We need to place a Tell Target Action in the button to jump to the Outline section on the main Timeline, which starts at Frame 31. We'll use the label outback as jumping point to this particular section of the main Timeline.



Within the outline Movie Clip, we place a Tell Target Action in the button to target the section labeled outback on the main Timeline.

When the user {Clicks} the outline button, the movie jumps to the label outback on the main Timeline and proceeds to play each of the Movie Clips in that section.

Open the Structure32.fla file in the Chapter11 folder if you haven't added the Tell Target Actions yourself.

Finally, to complete the introductory animation to this particular Flash movie, we need to place a Tell Target Action in the stripe Symbol to activate the outline Movie Clip at the label outlinebutton on the Main Timeline as well as placing an additional Tell Target Action to activate the contact Movie Clip at the label contact button.



There are three Actions in the Actions Layer of the stripes Symbol.

At Frame 23, outlinebutton is targeted on the main Timeline.
At Frame 25, contact button is targeted on the main Timeline. These Tell Target Actions will activate the outline and contact Movie Clips. At Frame 26, there is a Stop Action to prevent the stripes animation from looping.

Within the outline Movie Clip is an Instance of the Button outline. We need to place a Tell Target Action in the button to jump to the Outline section on the main Timeline, which starts at Frame 31. We'll use the label outback as jumping point to this particular section of the main Timeline.



Within the outline Movie Clip, we place a Tell Target Action in the button to target the section labeled outback on the main Timeline.

When the user {Clicks} the outline button, the movie jumps to the label outback on the main Timeline and proceeds to play each of the Movie Clips in that section.

Open the Structure32.fla file in the Chapter11 folder if you haven't added the Tell Target Actions yourself.

Summary

As a result of this chapter, you should be able to

- Designate a Movie Clip with a Unique Identification
- Use the Tell Target Command to control Movie Clips
- Understand the hierarchical structure of Multiple Timelines
- Use the Tell Target Command to structure Movies

CHAPTER 12

Interactivity:

Flash Forms and Linking to Web Pages



When building your web site, Flash is particularly suitable for the interface, for illustration purposes, for introductory animations, and the like. Standard HTML pages are particularly suitable for static and intensive text based content. Often, you will need to mix Flash and HTML together in a web site. This chapter covers building visually dynamic forms in Flash to gather information from your users, and linking to standard HTML pages outside of your Flash movies.

OBJECTIVES

Upon completing this section, you should be able to:

1. Build a Form in Flash
2. Understand Flash in HTML
3. Link to Web Pages from a Flash Movie
4. Use Flash to activate Email

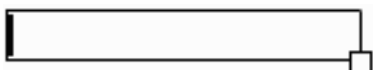
Forms in Flash

Building animated forms in Flash is a good way to encourage your users to fill out forms and talk about your site. To build animated forms, simply build animations as you normally do. The difference is, you must include a special type of text box, an interactive text field, and you must include a button that will send the variables and values of your form to a CGI script.

Text Fields

To include interactive text fields, meaning text fields in which your users can input information, {Click} on the Text Tool and

then {Click} on the Text Field modifier . {Click}, hold, and drag to create a text field to the width and height you desire. The font face and size that you have currently designated is the font face and size that users see when they input their information.



A text field.

Text Field Properties

Before you convert the text field into a Symbol, *{Right-Click}* the text field to set the text field's properties.

Text Field Properties

Variable:

Options:

- ☐ Draw border and background
- ☐ Password
- ☐ Multiline
- ☐ Word wrap
- ☐ Restrict text length to characters
- ☐ Disable editing
- ☐ Disable selection

Outlines:

- ☐ Do not include font outlines
- ☒ Include all font outlines
- ☐ Include only specified font outlines
 - ☐ Uppercase
 - ☐ Lowercase
 - ☐ Numbers
 - ☐ Punctuation
 - ☐ Characters:

Buttons: OK, Cancel, Help

Text field properties are divided into three sections: Variable, Options, and Outlines.

Variable

Input the variable name of the text field. The name you input into the Variable field becomes the variable name in the CGI script. The value of the variable will be input by the user. In this case we've named the variable name. This text field is meant to collect the user's name.

Options

Select from a variety of options that affect the type of text field you create and how it behaves:

Draw border and background

Check Draw border and background if you want Flash to automatically create a border around the text field and fill it with a background. In all cases, the border will be black and the background will be white. If you do not check this option, the text field will blend into the current color of your background. Simply place a rectangular shape around the text field.

Password

Checking this option will display asterisks * in the text field as users input their information.

Multiline

Checking Multiline enables the user to press the [RETURN] key to input multiple lines of text.

Word Wrap

When Multiline is checked, check Word Wrap to enable automatic wrapping of text when the user comes to the edge of the text field.

Restrict text length to& characters

Input a value into this field to restrict the number of characters (letters) a user can input into the text field.

Disable Editing

Check this option if you want to prevent users from changing the contents of a text field. In Flash, you can also use text fields to update text dynamically. This is covered in the Learn Flash Programming volume.

Disable Selection

Check this option to prevent users from selecting text in a text field. Users will not be able to cut and paste the text for example.

Outlines

Because different systems and different operating systems have different fonts, the font you select for your text fields may not be available on users' systems. Selecting a font such as Times New Roman is probably the safest font you can choose.

However, if you do not wish to use other fonts for aesthetic reasons, you can include the font outlines with your Flash file so that the user will see the proper font. This increases the size of your Flash file. Because we have included outlines in our particular Flash Form, the size of our movie is 29kb. Without including outlines, the size of the movie is 8kb. A considerable size difference.

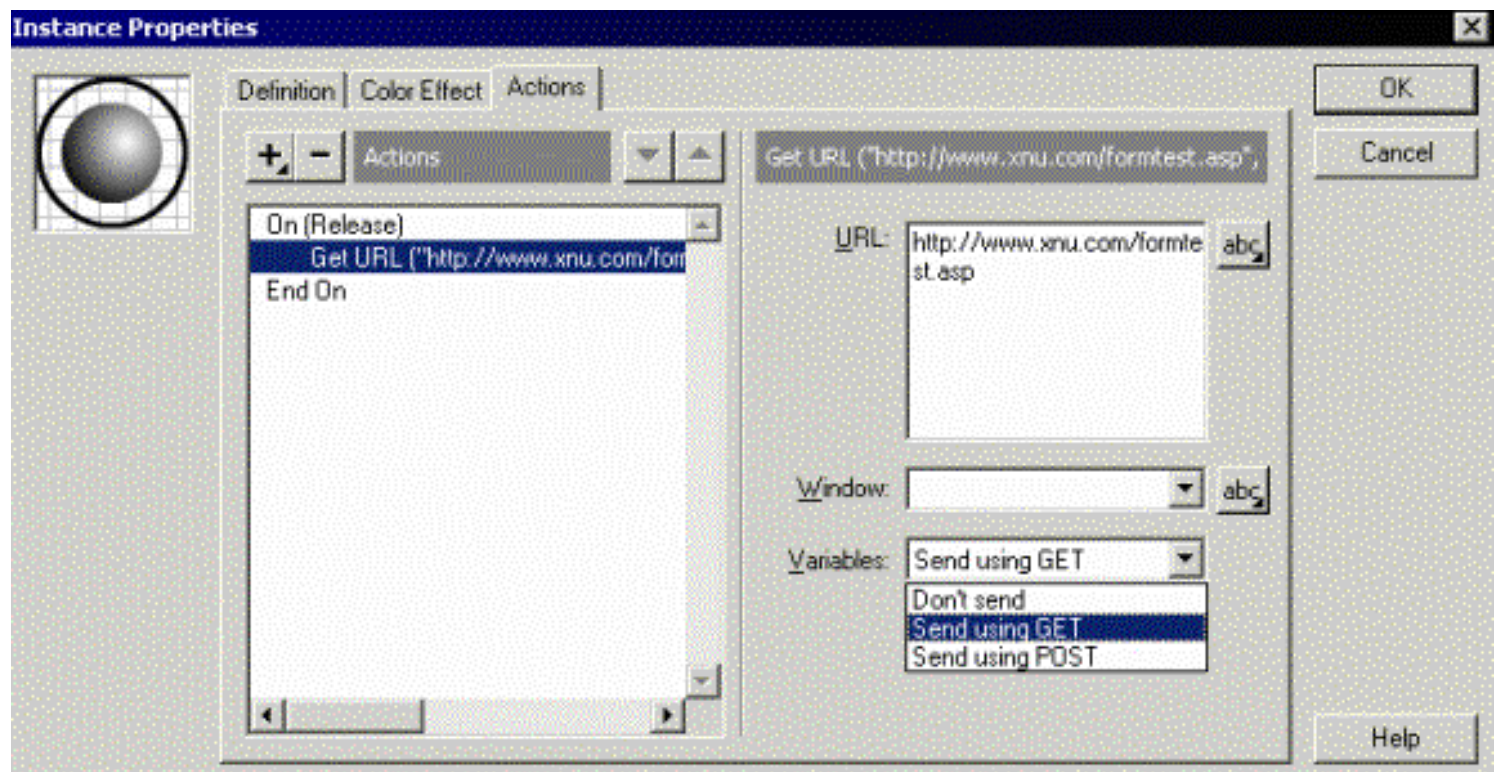
Check Include all font outlines to include the entire character set of the font.

Check Include only specified font outlines if you wish to reduce the amount of outlines you export. When this option is checked, you can choose to include, or not include, Uppercase, Lowercase, Numbers, Punctuation, or Characters by checking the appropriate checkbox.

Once you have created your text field, convert it to a Symbol. Use other text fields and standard Symbols to build your Form in Flash.

Processing the Form with a Button

Once you have built your Form, you'll need to send the variables and values to a CGI script. To do so, you must use a button. We attach a Get URL Action to the button. Simply {Double-Click} the button to add a button Action. Select Get URL. Input the path to the CGI script, in this case <http://www.xnu.com/formtest.asp>. In the Variables drop-down, select the method, either GET or POST. When you use the Get URL Action, your Flash movie must be referenced within an HTML document in a browser for the Get URL Action to function correctly.



When the user releases the mouse button the variables and values of the text fields are sent to the CGI script.

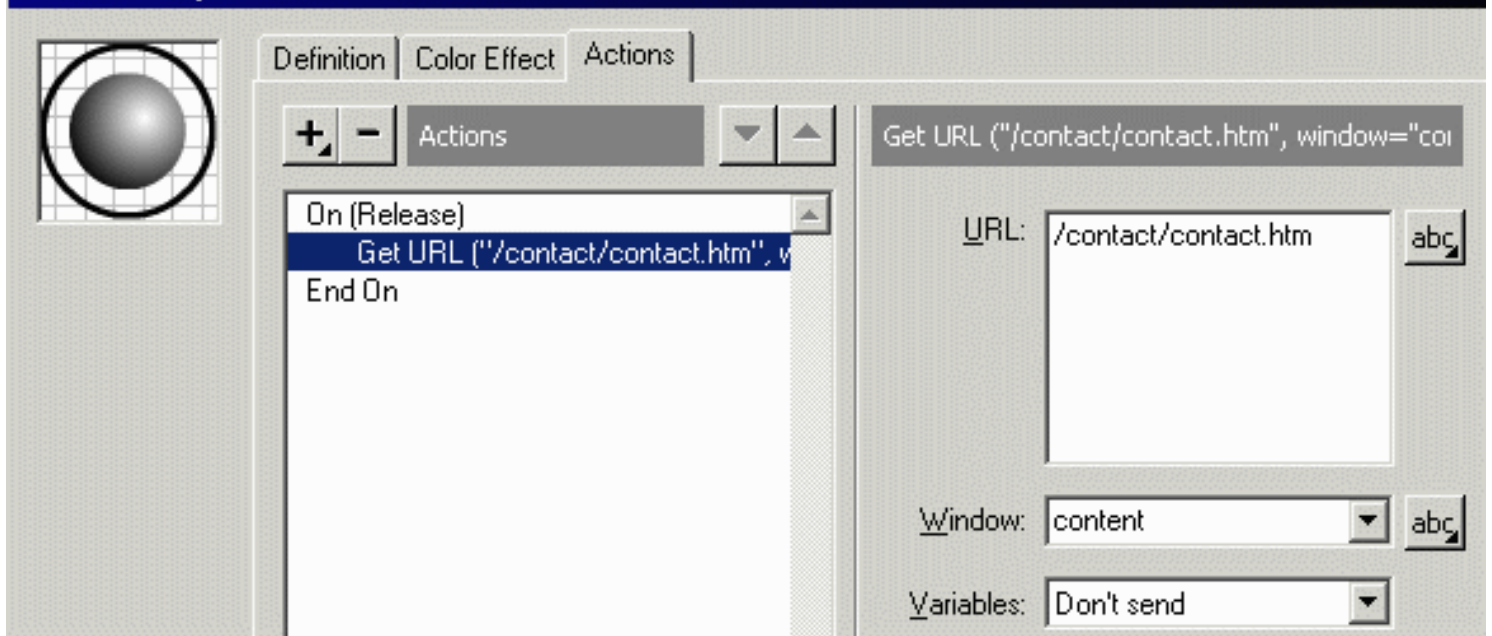
Open the file *form.htm* in the *Chapter12* folder in a browser.

We've build a form that requests users' name, email, phone number, and comment. When you {Click} the send button, the CGI script returns the name of each variable and the value you have input.

Flash in HTML Documents: Using Get URL

You place Flash movies in HTML documents in a similar manner to images, in that you link to the Flash movie (always the .swf files, not the .fla production files) within your web site. We cover placing Flash in HTML documents in the Preloaders and Publishing chapter. At this point, it's important to know that to use the Get URL Action, the Flash movie needs to be in an HTML page in a web browser. It can be the only content in the page, or it can be mixed with regular HTML content. Within the Flash movie, we can use a button to link to another HTML page that may or may not include Flash content. To do so, we simply {Double-Click} the button and select the Get URL Action. Input the path to the HTML document or other web site in the URL field. If you're using Frames, input the Frame name in the Window field. When users {Click} the button, the HTML page is loaded.

Instance Properties



When users **{Click}** this button, the page `contact.htm` is loaded into the browser and placed in the content frame.

Using Get URL for Email

To make a button open an email client, simply input <mailto:emailaddress@location.com> into the URL field with the Get URL Action.

Exercises

1. Open the file `Form.fla` from the Chapter12 folder.
2. Preview the movie.
3. Make a new movie in which you will build your own form.
4. Build a form animation with as many text fields as you want.
5. Create three buttons. One to send the form information, a second to link to your favorite web site, and a third to link to your own email.
6. Save the file as `MyForm.fla`.
7. Preview the movie you have just built. You will use it later in a HTML file.

Review Questions

1. How do you create a text field in Flash?
2. How do you designate variable names for text fields?
3. What do you have to designate in the Variables drop-down menu when using the Get URL Action to send variables to a script?
4. What function does the Window field serve when using the Get URL Action?

Summary

As a result of this chapter, you should be able to

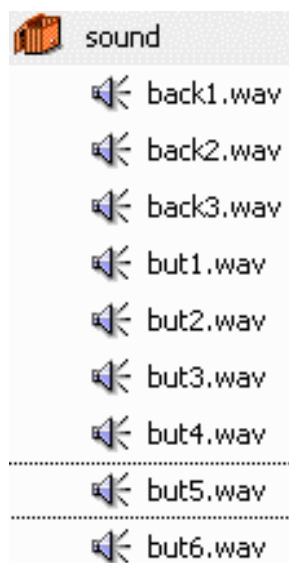
- Build a Form in Flash
- Understand Flash in HTML
- Link to Web Pages from a Flash Movie
- Use Flash to activate Email



Importing Sound Files into Flash

Flash does not create sound files, rather, you create sound with a sound program or acquire them from a sound collection. Flash imports Wave files (.wav) for Windows, and AIFF files for the Macintosh. To import a sound file, select File/Import from the main menu. Navigate to the folder where you've stored your sound file(s) and {Double-Click} the file you wish to import.

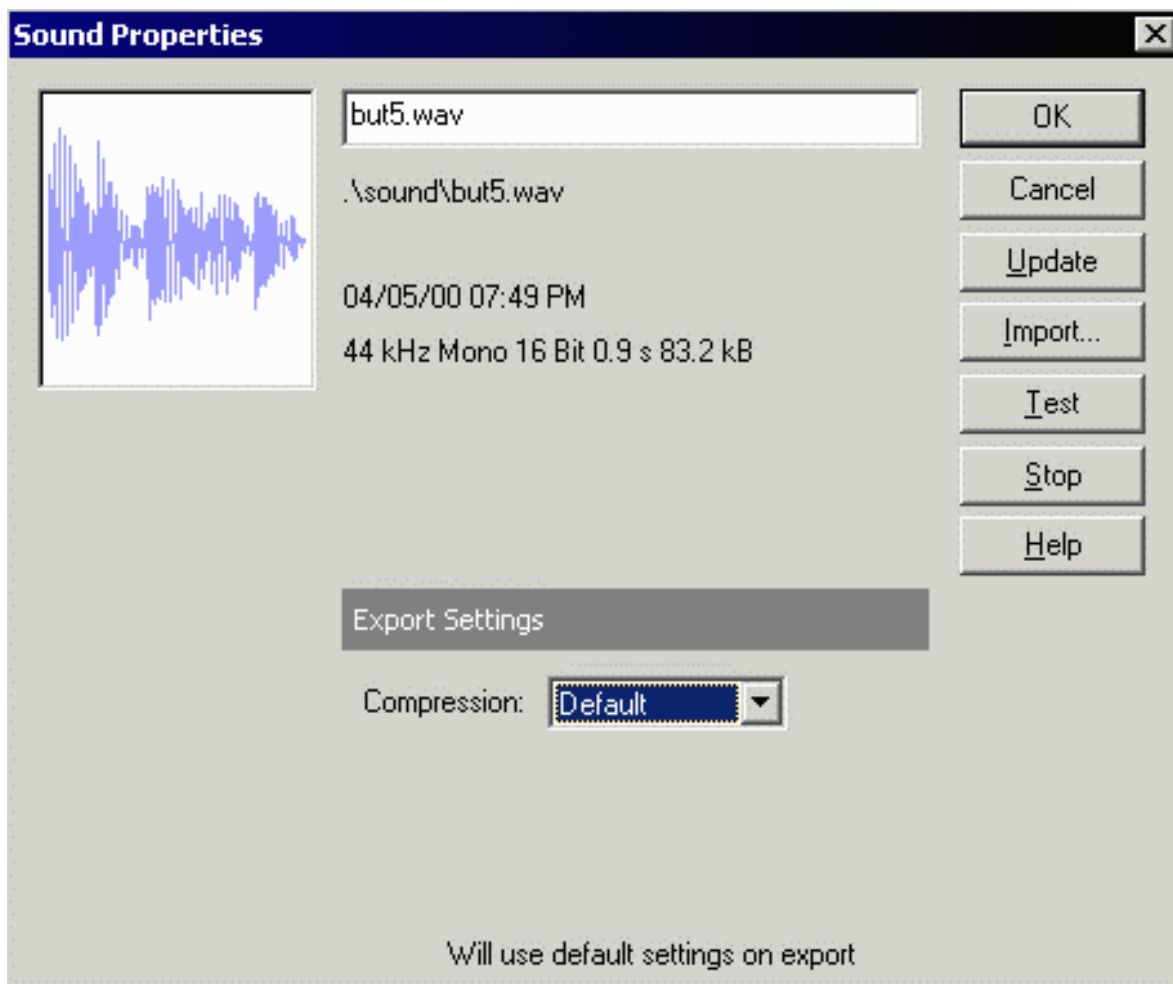
Sound files are not placed on the Stage. When you import sound files, they are placed in the Library.



A collection of sound files in the Library.

Setting Sound Compression

Before you use sound in Flash, you should set the compression settings. {Right-Click} the sound file in the Library and select Properties. The Sound Properties dialog box appears.



The name of the sound file, the path, the date it was created, the sampling rate, the bit rate, the duration, and the file size are displayed at the top of the dialog box.

Update

If you modify the sound file outside of Flash and resave it to the same location with the same filename, you can *{Click}* Update to reload the modified version of the file.

Import

{Click} Import to replace the existing sound file with a different sound file.

Test

{Click} Test to test the compression settings you apply to the sound using the Compression control in Export Settings.

Stop

{Click} Stop to stop a sound from playing when you have *{Clicked}* Test.

Export Settings Compression

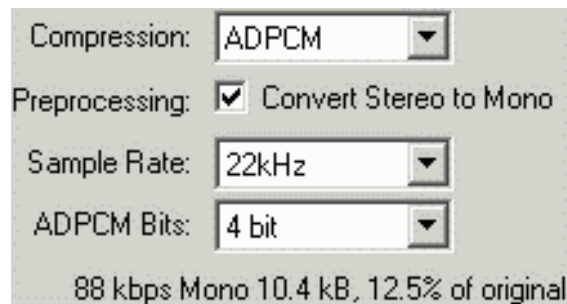
Use the Compression drop-down menu to compress the sound file, making it suitable for download.

Default

Leaving Compression at Default causes Flash to rely on the default sound settings you can set in Flash's Publish Settings, covered in a later chapter.

ADPCM

This compression method is best suited to small, quick sounds, such as clicking sounds you might attach to buttons.



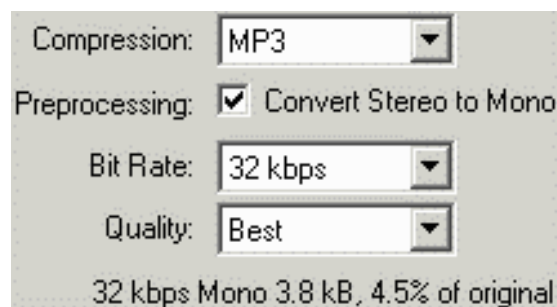
As you attempt different compression settings, Flash details the affect on the size of the sound by comparing it to the original imported size of the sound file. Its very important to *{Click}* Test to hear the affect your compression settings have on the sound.

For ADPCM, a higher Sample Rate increases sound quality and size. Similarly the higher the ADPCM Bits rate, the higher the sound quality and size.

Convert from Stereo to Mono also reduces the size.

MP3

This compression method is best suited to music files and loops. It is the most efficient compression sound method.



The higher the Bit Rate the larger the size. Similarly, the higher the Quality the larger the size.

Convert from Stereo to Mono also reduces the size.

Raw

Raw does not compress the sound. However, you can still reduce the size of the sound file by reducing the Sample Rate and converting stereo to mono. This will affect the quality of the sound.

Compression: Raw

Preprocessing: ☒ Convert Stereo to Mono

Sample Rate: 22kHz

Using Sound in Flash

To use a sound sample you have imported, create a Layer in the Timeline for sound. Place a key frame at the point you wish the sound to begin. Sounds are attached to key frames, similar in the manner that Actions are attached to key frames. {Double-Click} the key frame and select the Sound Tab from Frame Properties.

Frame Properties

Label

Sound

Actions

Tweening

Sound: None

0.51.0

No sound selected.

Effect: None

Sync: Event

Loops: 0

OK

Cancel

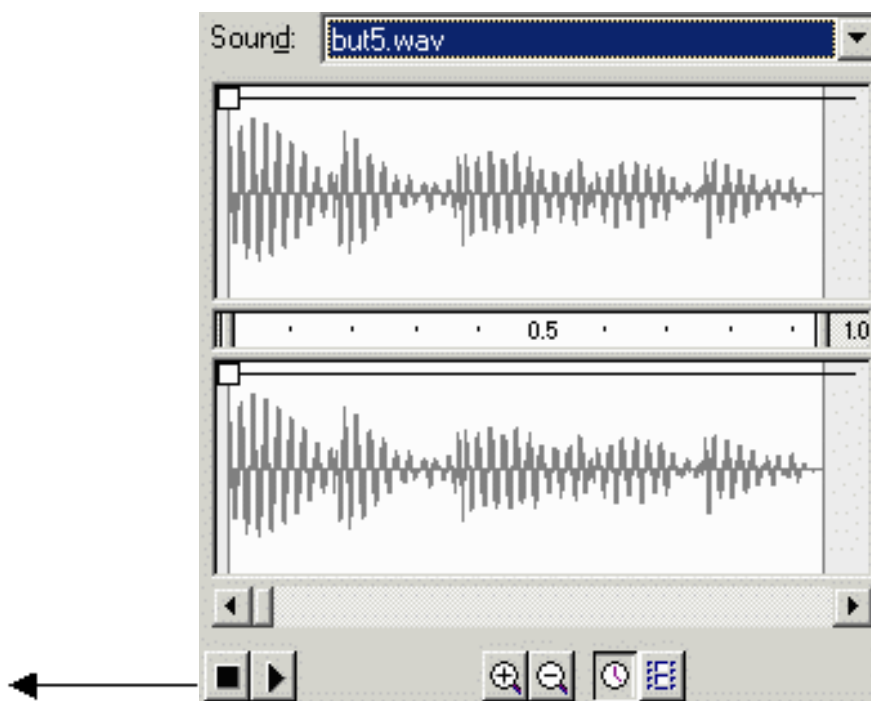
Help

Assigning and Modifying Sound

To assign a particular sound to this key frame, select the sound file name from the Sound drop-down menu. The waveform pattern of the sound is displayed in each channel.

Envelope Line/Handles

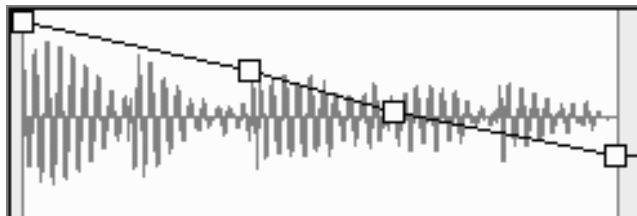
Time In/Out Controls



Stop Play Zoom In/Out Time Seconds/Frames

At the bottom of the window, from left to right, are Stop, Play, Zoom In, Zoom Out, Time in Seconds, and Time in Frames buttons.

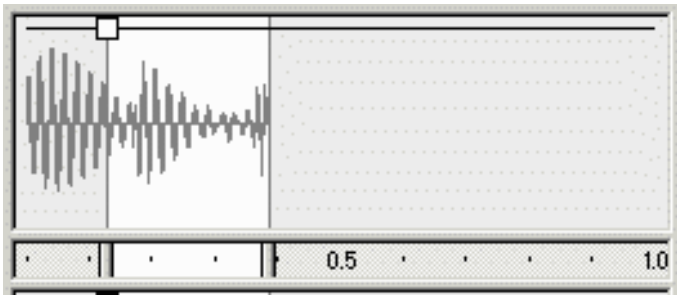
If you wish to affect the volume of the sound sample, {Click} on the Envelope Lines above each channel to add more Envelope Handles. Drag the handles in both channels to increase or decrease volume.



Envelope Handles are added to the Envelope Line by {Clicking} on the Line.

To remove Envelope Handles, simply drag them to top of, and beyond the channel box.

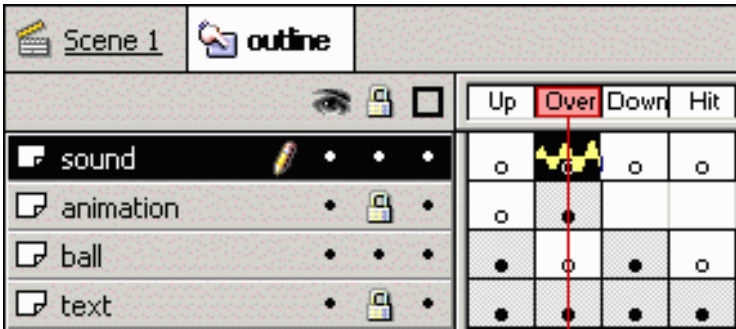
To modify where the sound sample starts and stops, drag the Time In and/or Time Out controls. The white area of the channel box is played, the gray area is not.



The sound starts and ends at different points by dragging the Time In/Out controls.

Use these controls sparingly to perform small modifications. Major modifications should be performed with a full featured sound application.

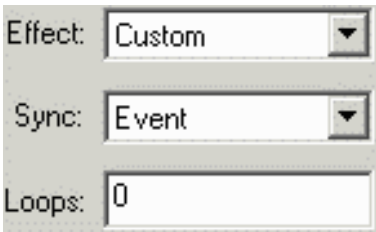
When the Marker passes over the particular key frame, the sound will play. For example, you could place a particular sound in the Over state of a Button.



A sound sample is placed in the Over state of the outline Button.

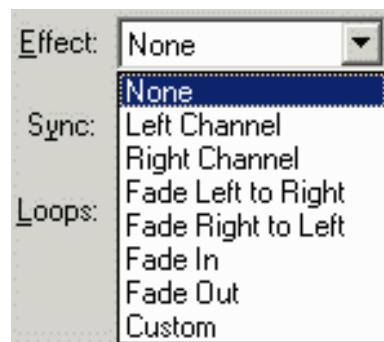
Setting Sound Behavior

On the right side of the Sound controls are three categories that affect how the sound behaves in your Flash movie.



Effect

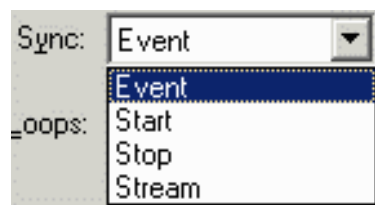
To add particular effects to a sound, select from Left Channel, Right Channel, Fade Left to Right, Fade Right to Left, Fade In, Fade Out, and Custom.



If you modify the volume or start and stop points of the sound, select Custom.

Sync

Select from Event, Start, Stop, or Stream.



Event

Event sounds cannot play until the entire sound is downloaded into memory from the web. If you use Preloaders to ensure that the entire contents of your movie are downloaded before beginning to play it, Event is the preferred method. Preloaders are covered in a later chapter.

Start

If you want a sound that is already playing to start again select Start.

Stop

Select Stop to stop a sound that is currently playing. You might want to stop a looped sound or a long sound at a particular point in the movie. Simply place a key frame at the particular point in the Timeline, select the name of the sound and select Stop from Sync.

Stream

Select Stream if you wish to force the animation to keep up with the sound. If the animation falls behind, frames are skipped.

Loops

Flash does not include a Loop checkbox to automatically loop the sound forever. To loop a sound continuously, enter a value of 9999 in the Loop field. Input other values if you want the sound to loop a given amount of time. A value of 0 will play the sound through once if it is an Event sound.

Exercise

1. Open the file Sound.fla from the Chapter13 folder.
2. Preview the movie. Notice that a sound has been added to the Outline button at the Over state.
3. In the sound folder of the Library is a collection of sounds.
4. Examine each sound file's properties and play with the compression settings.
5. Add sounds to the XNUanim Movie Clip, as well as the back button, the contact button, and the module buttons.
6. Note how the use of more sounds and variations in compression settings affect the file size of the .swf file.

Review Questions

1. When you import sound into Flash, where is the sound placed?
2. How do you set the compression level for sound?
3. How would you place a sound on the Down state of a Button?
4. What is the best method of compression available in Flash?

Summary

As a result of this chapter, you should be able to

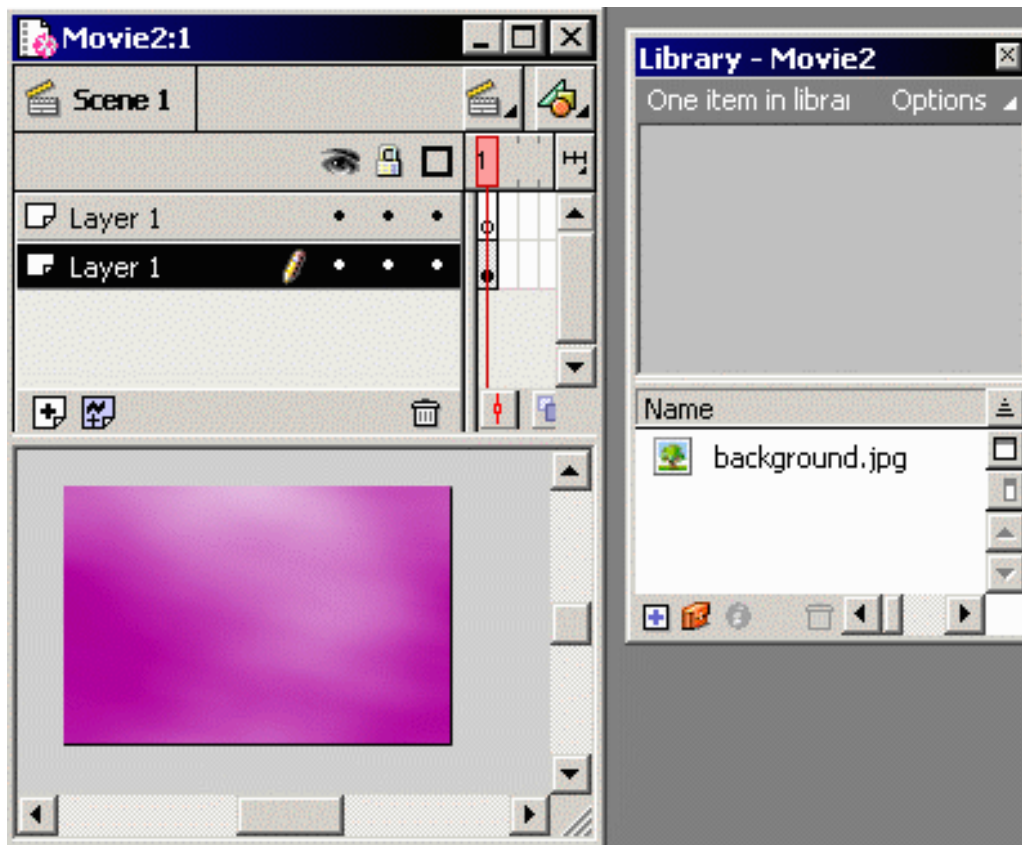
- Import Sound files into Flash
- Set Compression for Sound
- Use Sound in Flash



Importing Bitmap Images

Flash can import a variety of bitmap file types, including JPEG, GIF, PNG, and BMP (with Windows) and PICT (with Macintosh). Flash will compress bitmaps that you import itself, so in all cases, we don't recommend compressing an image before importing it into Flash.

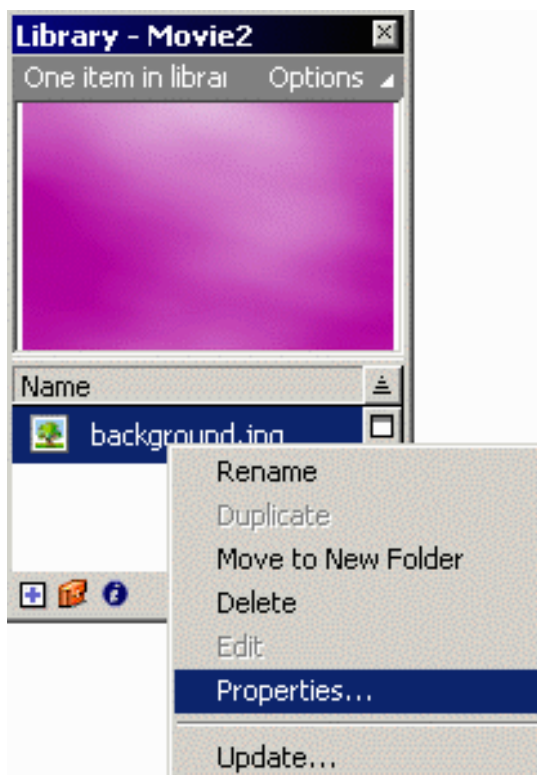
To import an image, select File/Import. Select the appropriate file format or All Formats for file type and *{Double-Click}* the bitmap file. The bitmap is placed on the Stage on the Overlay Level. It is also placed into the Library as a bitmap, not a Symbol. Converting it to a Symbol permits you to apply the standard Instance Properties and effects to the bitmap. Keep in mind that animating bitmaps takes significantly more processing power than animating vectors.



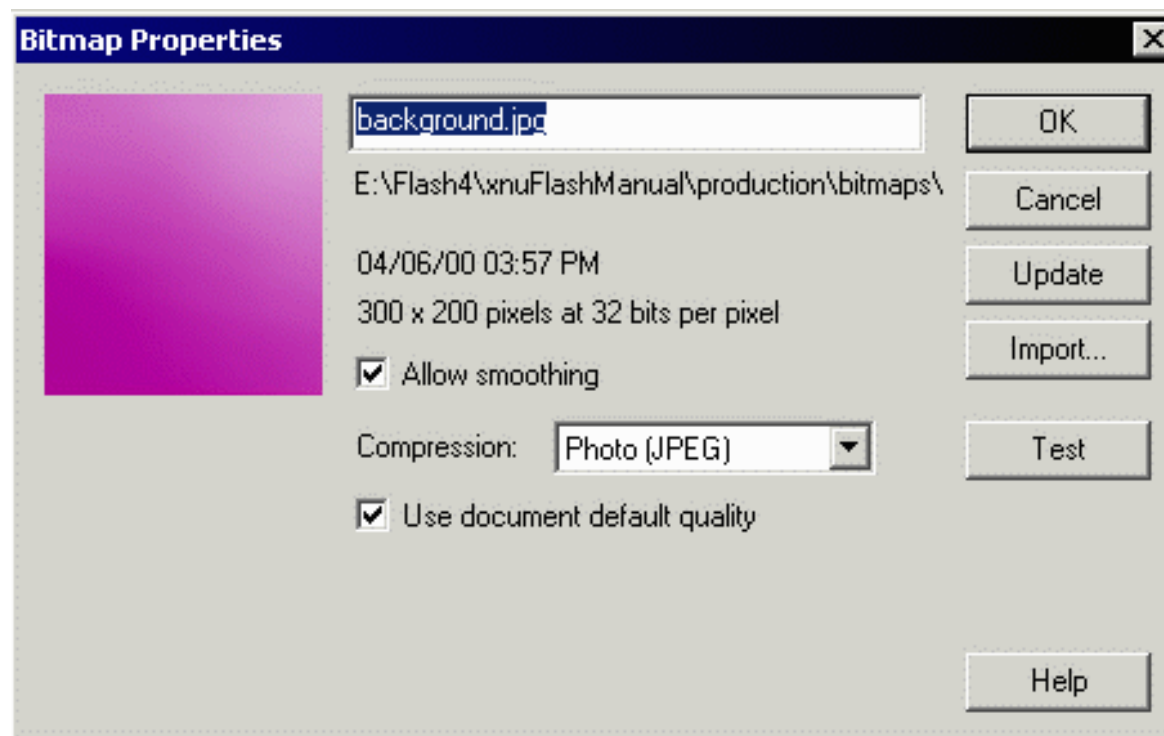
The bitmap is placed on the Stage on the Overlay Level. It is also placed in the Library, but it is not a Symbol. You have to convert it to a Symbol if you wish to use Instance Properties and effects on the bitmap.

Setting Bitmap Properties

Before you use the bitmap you should set the Bitmap Properties which affect the way the bitmap is displayed in your Flash movie. {Right-Click} the bitmap in the Library and select Properties.



The Bitmap Properties dialog box appears.



Update

{Click} to reload the bitmap file. If you have modified the bitmap file outside of Flash, in a program like Photoshop or Fireworks for example. Simply save it with the same name and reload the file with Update. The path and the date the original image was imported is displayed beneath the filename.

Import

You can import an alternative image by *{Clicking}* the Import button and navigating to a bitmap file.

Allow Smoothing

Check this option to enable anti-aliasing on the bitmap. The edges will be smoothed.

Compression

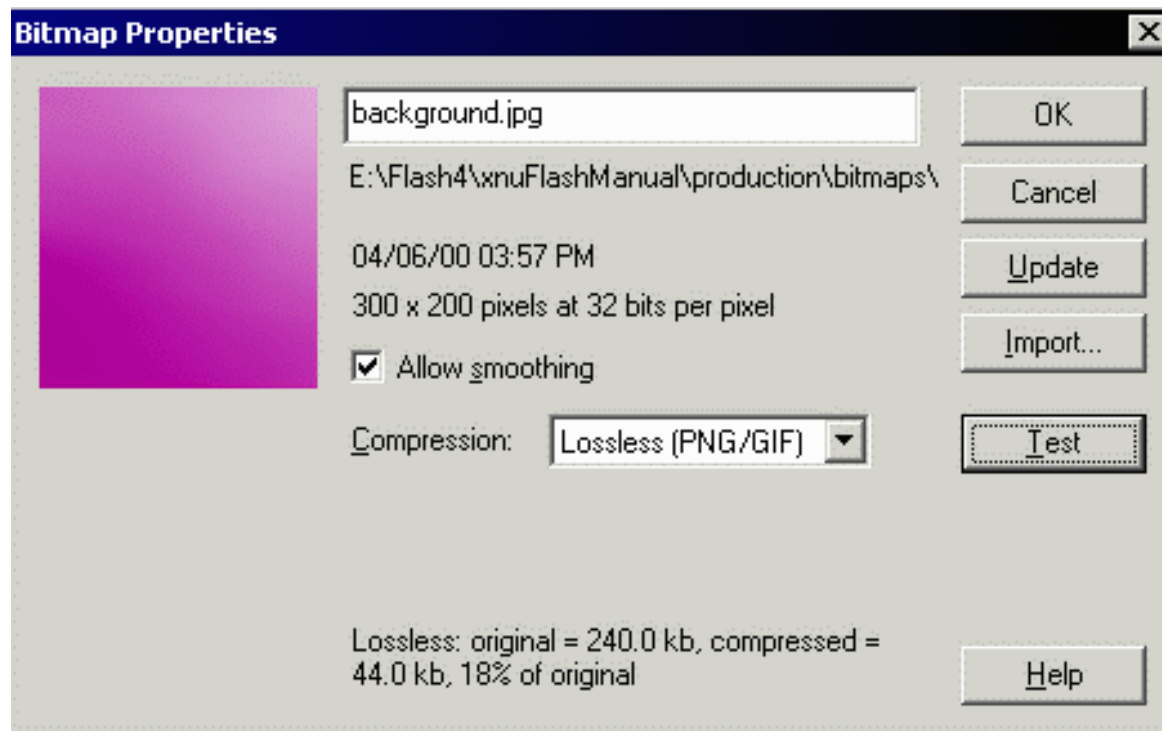
Select from Photo (JPEG) or Lossless (PNG/GIF). Lossless generally produces a higher quality image but also a larger image in terms of file size.

Use Document Default Quality

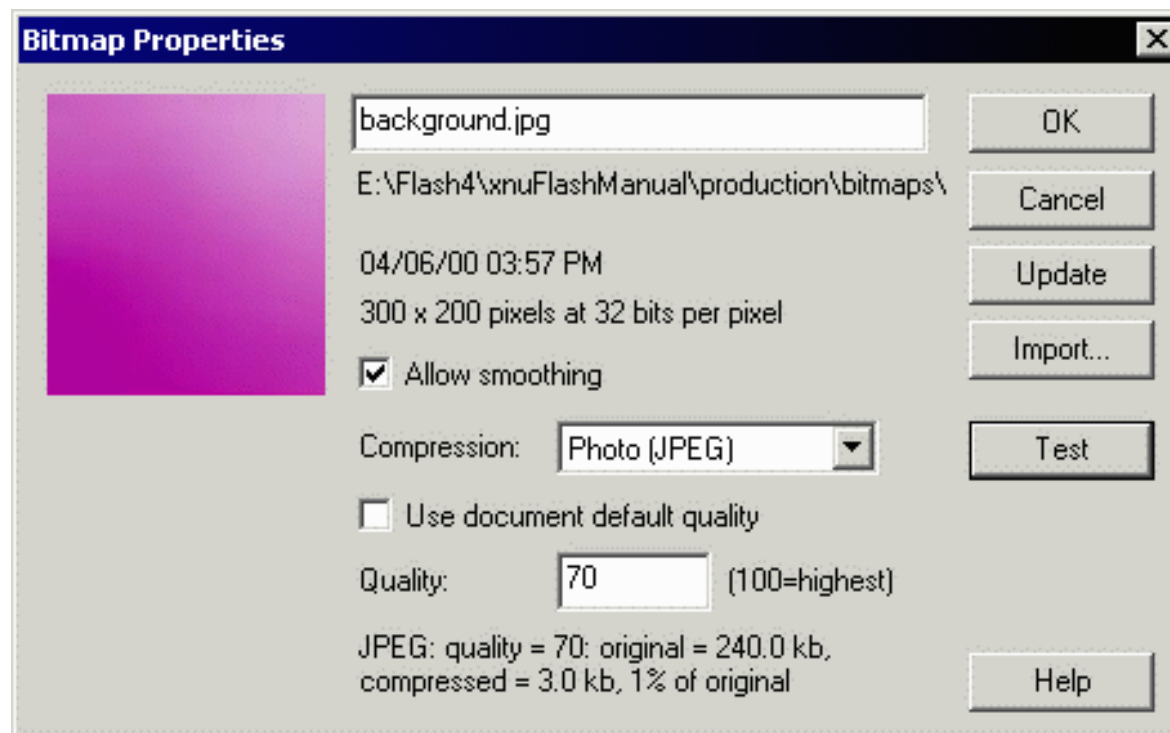
You can set a default quality for JPEG compression in Flash's Publish Settings (covered in the next chapter). Uncheck this to set the compression for the particular image you are using.

Test

{Clicking} Test produces a comparison of the original file size to the size of the image with compression applied to it.



With Lossless (PNG/GIF) compression, the image size is reduced to 44kb from 240kb, 18% of the original size.



With Photo (JPEG) compression set to 70, the image size is reduced to 3kb from 240kb, 1% of the original size.

Importing Vector Images from other Applications

There are a number of native vector applications such as Macromedia's Freehand, Adobe's Illustrator, and Corel's Draw. Each of these applications can be used to create vector images for print and the web. You may prefer to create advanced vector based images in these dedicated programs, or you may need to import a logo from a client. Each of these programs can save files in two standard vector file types, Adobe Illustrator (.ai), and the Adobe Encapsulated Postscript (.eps). Fortunately, Flash can import Illustrator and EPS files.

To import Illustrator or EPS files, select File/Import from the main menu. Select the appropriate file type for format and {Double-Click} the file. The vector is placed on the Stage at the Overlay Level. If vector files contain Layers, Layers with the separate content may be created in the main Timeline. Simply select all the content across Layers if you need to, and convert the imported vector content to a Symbol.

Exercise

1. Create a new movie at 300 pixels wide by 200 pixels high.
2. Import the bitmap file background.jpg from the Chapter14 folder.
3. Examine the properties of the bitmap. Try out different compression levels.
4. Import the vector file spiralpath.ai from the Chapter14 folder. Paths for animation guides are often created in vector programs such as Macromedia's Freehand.

Review Questions

1. When vectors or bitmaps from other applications are imported into Flash, what level are they imported to?
2. How do you set the compression quality of bitmap images?
3. How do you view the affect of the compression settings you've applied to a bitmap image?

Summary

As a result of this chapter, you should be able to:

- Understand how to import bitmap images
- Set Bitmap Properties
- Import Vector Images from other Applications



The Preloader

Although Flash supports streaming over the Internet, streaming at this point in time, because of bandwidth variables, does not always work properly. When a Flash movie streams, it begins to play the movie once it has determined there is enough of the movie's content in memory. As it plays, it continues to download the rest of the movie. However, if the movie continues to play to a point where content has not yet been downloaded, the animation will pause. To prevent this event, you can ensure that the entire contents of the movie are downloaded by building a Preloader. Preloaders check to see if a particular frame is loaded or if all the frames are loaded before beginning to play the movie.

There are two methods, both using Actions. One method checks to see if all the frames are loaded in the movie, and the other checks to see if a particular frame is loaded. In both cases, the Actions are built to loop until the condition of all frames or a particular frame loaded is true.

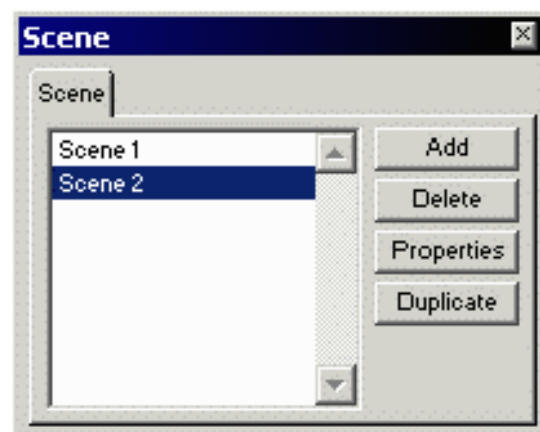
Planning a Preloader

Before you build a Preloader, you should create a looping Movie Clip to inform the user that the movie is in the process of loading. A common approach is to display pulsing text with the message &loading movie or loading. Once you've created the Movie Clip, the usual process is to add a Scene to your movie in which you place the Movie Clip and the Preloader Action.

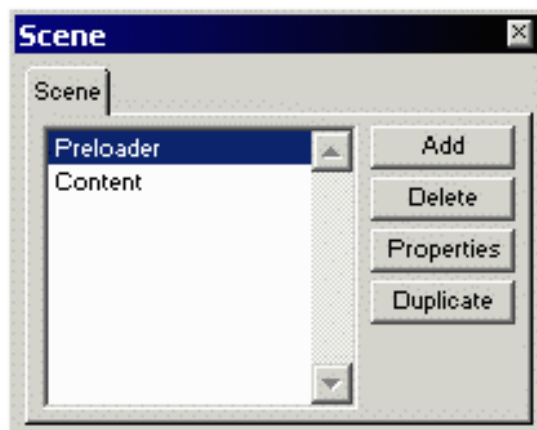
Creating Scenes

To add a scene select Insert/Scene from the main menu. A new scene is created in your movie. When a new Scene is created, nothing from the previous Scene is brought over to the new Scene. If you are using particular Instances through your movie, you would have to copy the Layers from one Scene to another, which is why, generally, most Flash developers use only once scene and build their animations with Movie Clips. However, Scenes are very convenient for Preloaders, because the visual content of Preloaders often differs from the content of the movie.

When you create a new Scene, by default, it's placed after the current Scene. To change the order of Scenes, select Window/Inspectors/Scene from the main menu. The Scene Inspector appears.



Because we've just added a Scene, our movie now consists of two Scenes, 1 and 2. Scene 1 is where our content is. Scene 2 is where we are going to place our Preloader. Simply drag Scene 2 above Scene 1 in the Scene Inspector to rearrange the Scene order. You might want to rename each Scene as well, such as Preloader and Content for example. To rename a Scene, *{Double-Click}* the Scene name in the Scene Inspector.



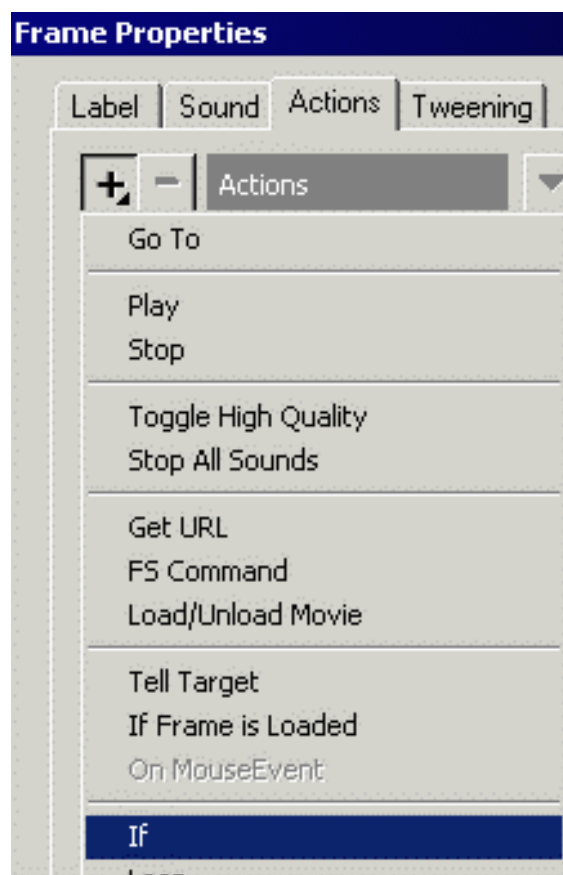
Scenes are rearranged and named appropriately.

{Clicking} on a Scene name places you within that particular Scene. Now you're ready to build your Preloader.

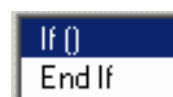
Using If (`_framesloaded`)

The If (`_framesloaded`) Action checks to see if all the frames in the movie have been loaded. If they have, you can instruct Flash to Go to and Play the next Scene, or a particular frame. If all the frames have not been loaded, you can instruct Flash to continue to check if they have been, placing the movie in a loop until the condition of all the frames being loaded is true.

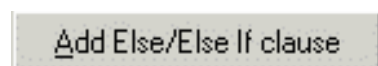
The first step, is to place the Movie Clip you've made for the Preloader in the first frame of the first Layer. Next, *{Double-Click}* the key frame to add a If (`_framesloaded`) Action. Select If from the Actions menu.



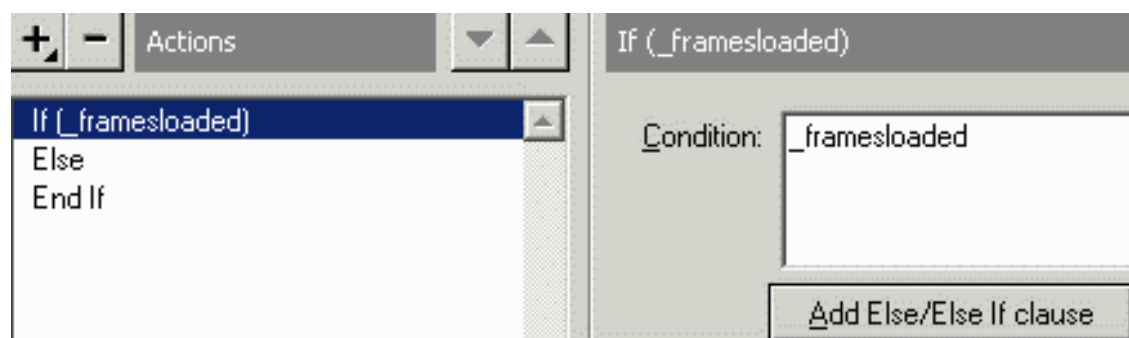
You are presented with an If () and End If statement.



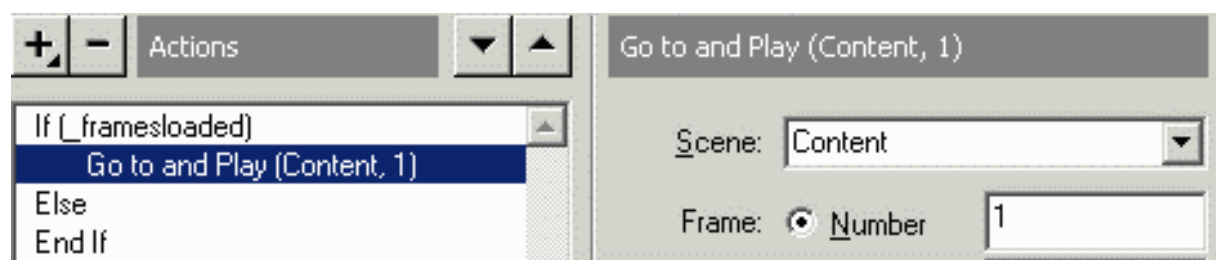
{Click} the Add Else/Else If clause button to add an Else statement to our syntax.



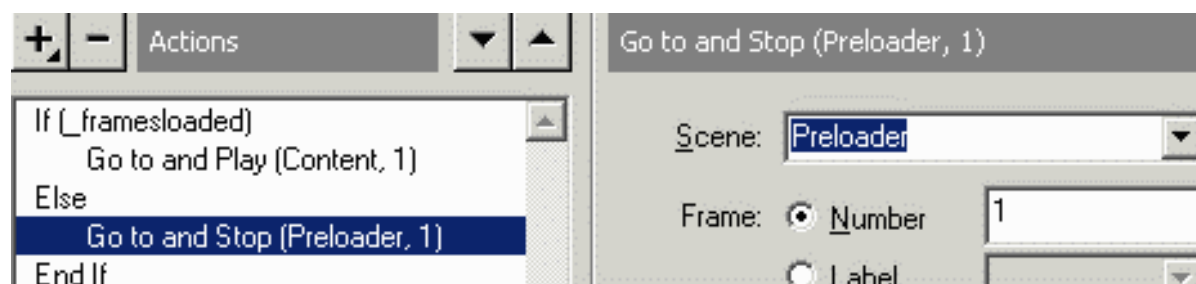
In the Condition field, input `_framesloaded` (without quotation marks).



Next select the Go to Action, select the scene you wish to jump to when all the frames have been loaded (in this case Content), and check Go to and Play to ensure the content begins to play.



If the all the frames have not been loaded, we need this Action to do something else, namely, to check again until the condition If (_framesloaded) is true. We select the Else statement and add another Go to statement, this time to go to the current frame so that the condition is checked until true. Effectively, we are creating a loop. When the If (_framesloaded) condition is true, we jump out of the loop and the movie proceeds.

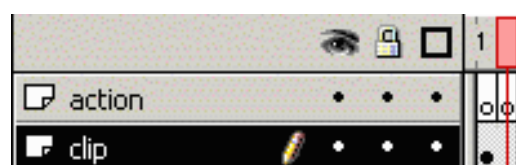


Add a Go to and Stop to the Else statement, instructing Flash to go back to, or remain on the current frame in the current scene until the condition If (_framesloaded) is true.

Using If Frame is Loaded

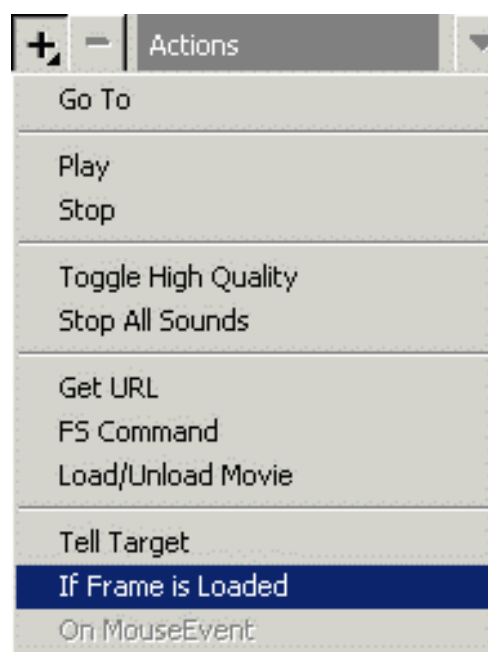
Using the If Frame is Loaded Action requires the use of two key frames. It differs from the If (_framesloaded) Action in that you can check to see if a specific frame is loaded. Again, we create a looping Movie Clip to inform users the movie is downloading.

Place the Movie Clip in a Layer and extend its range to two frames. Create a new Layer for Actions and place blank key frames in Frames 1 and 2.

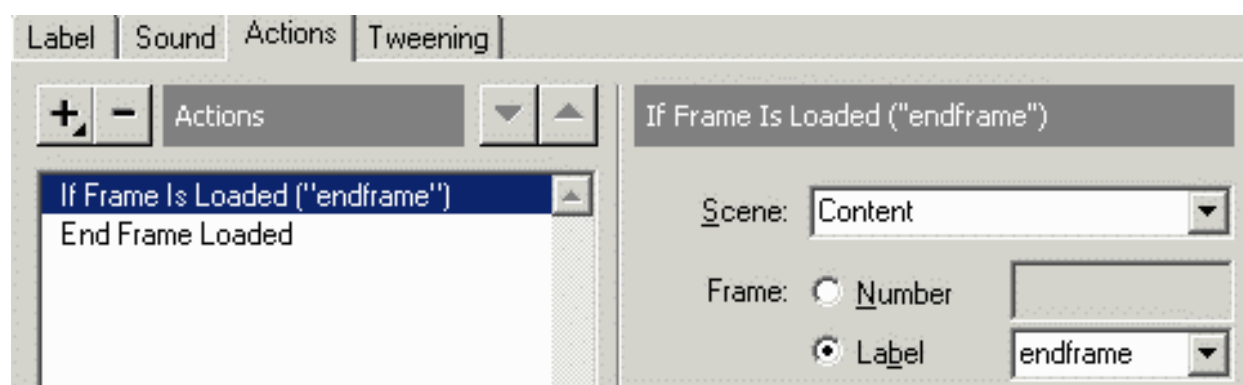


The Movie Clip is placed on one Layer, the Actions in another Layer.

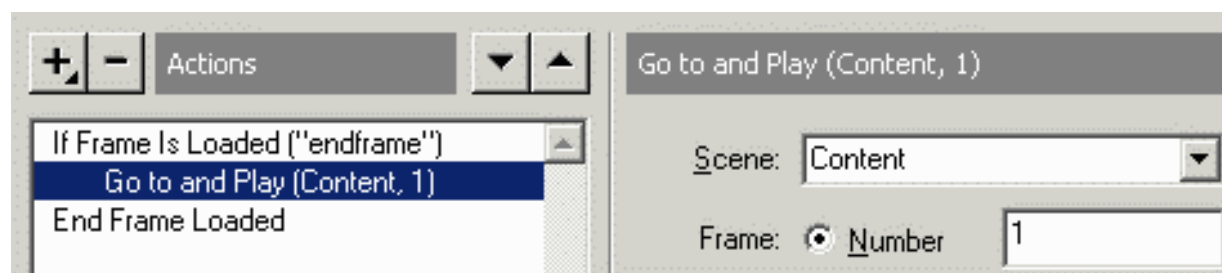
We add the If Frame is Loaded Action to the first key frame on the Actions Layer. {Double-Click} the key frame and select the If Frame is Loaded Action.



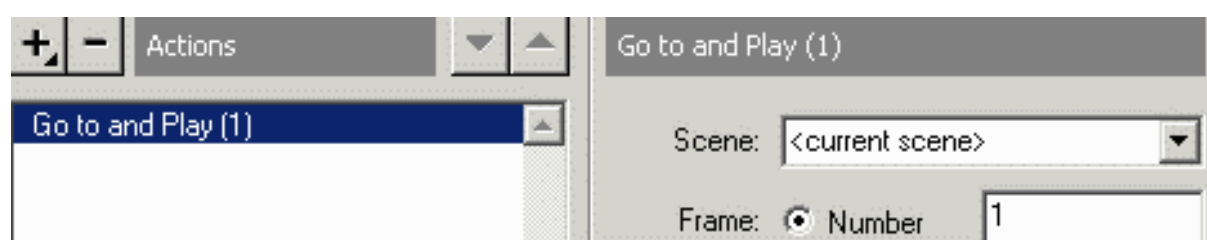
Select the Scene in which the frame you are checking for is in, and then select either the frame number or frame label. In this case, we've placed a key frame at the end of our movie labeled endframe in the Content Scene.



Next we need to tell Flash what to do if the particular Frame is loaded. We select a Go to Action and tell it to play the first frame of the Scene Content.



If the condition is not true, if endframe is not yet loaded, then Flash ignores the Go to and Play statement and proceeds to the next frame in the Timeline. We need to place a Go to Action in the second key frame of the Actions Layer to go back to Frame 1 so that the condition can be checked again. When the condition is true, the movie will proceed to play, jumping out of the loop.



Preloaders are not necessary for every Flash movie. It will be up to you to

decide if your Flash movie needs a Preloader. The general rule is, the larger the file size, the more you might require a Preloader. Once you have created your Flash movies and created Preloaders where necessary you can publish your site to HTML.

Open the file Preload.fla from the Chapter15 folder and preview the movie.

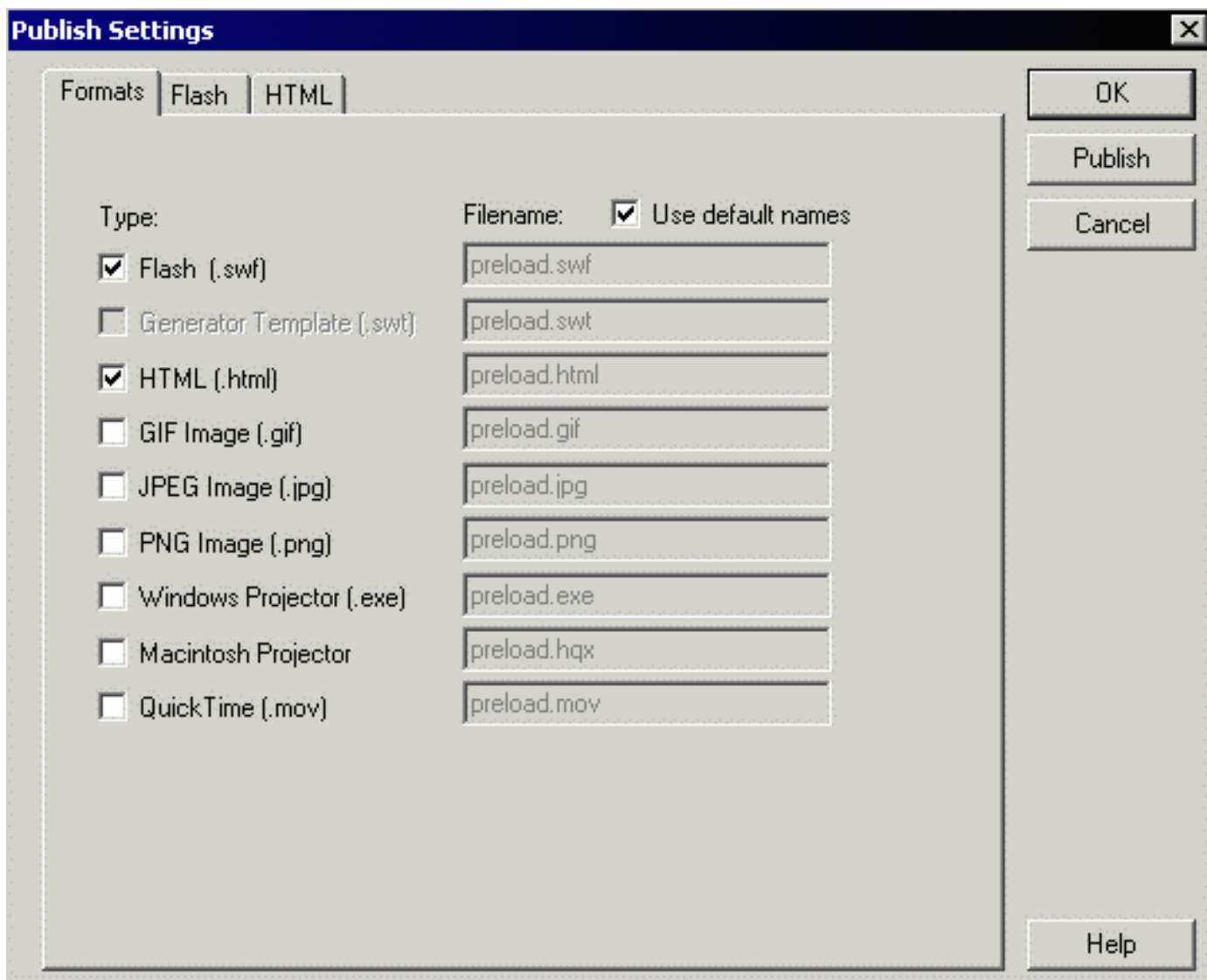
To preview your Preloader, preview the movie as you normally would. Because we are previewing the movie off of a hard drive, the load time is instant. In order to simulate downloading from the Internet, press [CTRL] + [ENTER] a second time.

Publishing to HTML

When you publish a Flash file to HTML, you publish the .swf file, which is the optimized version of your Flash movie. In order for users to see your movies on the web, your movies must be placed in a web page and users must have the Flash plug-in. There are three methods of publishing you can use. If you are familiar with HTML and prefer to code by hand, you can use the OBJECT and EMBED tags, together with the appropriate attributes and values to place your Flash movies in web pages. If you do not wish to code by hand, you can use Flash to publish the movie to an HTML page automatically, that you can later edit yourself or with a third party authoring system. Finally, if you use Macromedia's leading HTML authoring package, Dreamweaver, you can use it to publish the Flash movies you have made.

Using Flash to Publish to HTML

Flash can automatically generate both the net-ready Flash movie (the .swf file) and the HTML file the Flash file is placed in. To control how both the movie and HTML page is generated, select File/Publish Settings. The Publish Setting dialog box appears.

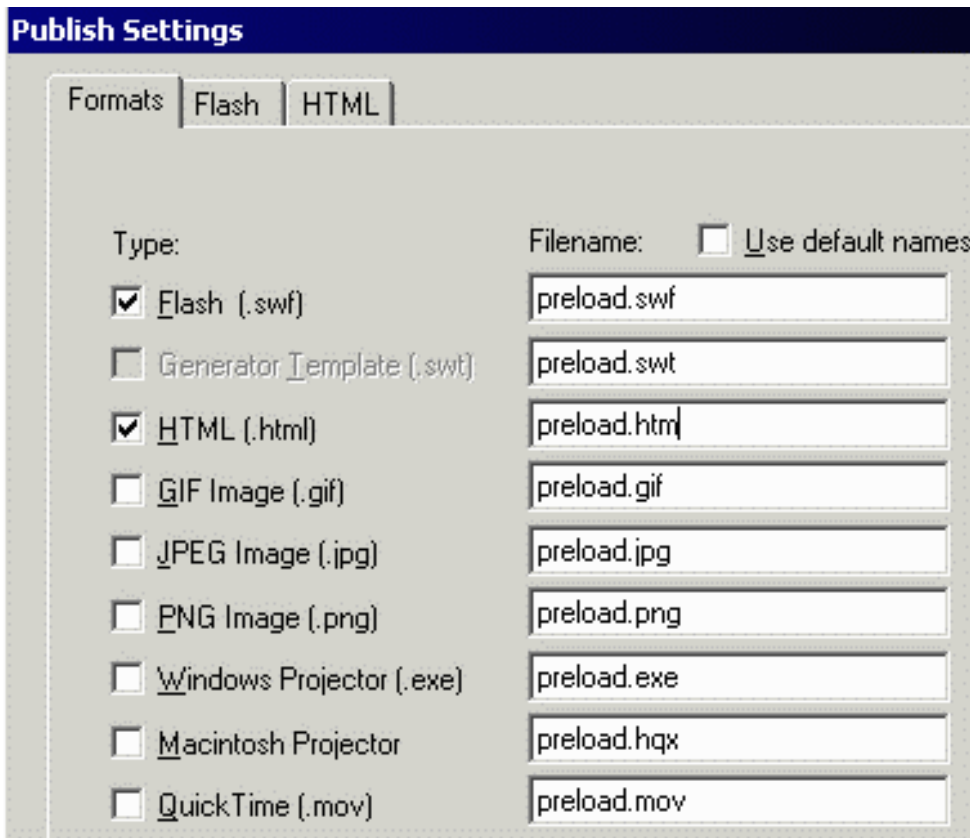


The Publish settings are divided into three categories, Formats, Flash, and HTML.

Formats

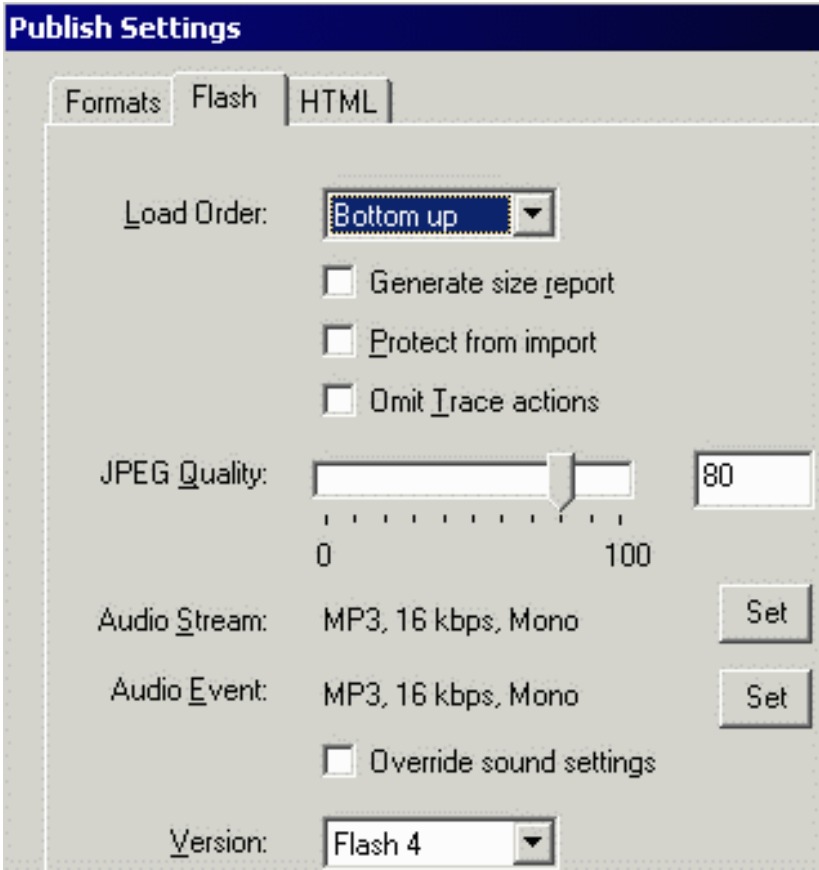
The Formats category controls which formats Flash will publish when you *{Click}* the Publish button. By default, Flash and HTML are checked. If you check additional formats, category tabs appear in Publish Settings (with the exception of the Windows and Macintosh projectors). *{Click}* the appropriate tab to modify settings.

If you uncheck the Filename: & Use default names checkbox, you can input your own custom names for the files Flash generates. For instance, many designers prefer to use the .htm extension, rather than the .html extension, which is the Flash default.



With Use default names unchecked, you can change the name and extensions of the files Flash generates.

Flash
 Select the Flash tab to control the settings for the .swf files that Flash creates when you {Click} the Publish button in Publish Settings, select Publish from File/Publish, or press [CTRL] + [ENTER] when previewing your movies.



Load Order

When a Flash movie streams, by default, the Layers are loaded from the bottom of the Timeline to the top of the Timeline. Usually, you'll place backgrounds on the bottom Layer, making Bottom up a good choice for Load Order. Of course, if you use Preloaders, this isn't an issue. You can also select Top down as an alternative Load Order.

Generate Size Report

Checking this option generates a text file (.txt) of the same name as the Flash file. The text file details the size of each frame in terms of bytes, as well as the size in bytes of Symbols, divided into shape and text categories.

Protect from Import

Check this option to prevent your Flash movies from being imported into Flash by other Flash users.

Omit Trace actions

Check this option to prevent Flash from using the Trace function. The Trace function generates a window in particular situations that reports errors and other content.

JPEG Quality

Use the slider to increase or decrease the JPEG compression that Flash applies to any bitmaps you've placed in your movie. The lower the value, the higher the compression and therefore, the smaller the file size. However, higher compression results in lower image quality.

Audio Stream

If you stream audio in your Flash movie, you can set compression settings for your streaming audio. These settings only apply to audio samples that you haven't set compression settings for individually.

Audio Event

You can set compression settings for your event audio. These settings only apply to audio samples that you haven't set compression settings for individually.

Override sound settings

Check this box to force the compression settings you've set with Audio Stream and Audio Event to override the settings you've made on an individual basis.

Version

Select the version of the Flash movie you wish to generate. You can generate Flash files from versions 1 to 4. Flash will warn you if you've used features in newer versions that aren't compatible with older formats.

HTML

Select the HTML tab to control the settings for the HTML page that Flash generates when you *{Click}* the Publish button in Publish Settings or select File/Publish from the main menu.

The screenshot shows the 'HTML' tab in the Publish Settings dialog. The 'Template' dropdown is set to 'Flash Only (Default)'. The 'Dimensions' dropdown is set to 'Match Movie'. The 'Width' is 500 and 'Height' is 300 pixels. Under 'Playback', 'Loop' and 'Display Menu' are checked. 'Quality' is set to 'High'. 'Window Mode' is set to 'Window'. 'HTML Alignment' is set to 'Default'. 'Scale' is set to 'Default (Show all)'. 'Flash Alignment' has 'Horizontal' set to 'Center' and 'Vertical' set to 'Center'. The 'Show Warning Messages' checkbox is checked.

Template

Flash contains HTML templates that control the content of the HTML generated when you Publish the movie. For details of the content, select the Template and *{Click}* the Info button. Flash Only is the default and is the preferred method of publishing. In all cases, when a Flash movie is included in HTML, Flash uses both the OBJECT and EMBED tag. OBJECT is Explorer specific, EMBED is Netscape specific. Most of the settings in the HTML section are parameters for both tags.

Dimensions

By default, Match Movie is selected for dimensions, so that the width and height of your Flash movie is set to match the width and height you set in Movie Properties. You can also select Pixels, and input values into the width and height fields in pixels, or you can select Percent, and input values into the width and height fields in terms of percentage of the browser window. Values of 100% will force the Flash movie to scale to the current size of the browser window.

Playback

Playback affects a number of parameters for the Flash movie.

Paused At Start

Checking prevents the Flash movie from automatically playing. To play the movie, you need to include a start or play button at the very beginning. By default, this parameter is left unchecked.

Display Menu

Uncheck Display Menu to prevent the option of a context-sensitive menu *{Right-Click or CTRL-Click}* for the Flash file. By default this option is checked, permitting users to zoom in and out, restart your Flash movie, and so on.

Loop

Checking this option forces Flash to automatically loop. Since we use Frame and Button Actions to control the flow of the Flash movie, it's a good idea to keep this option, which is the default option, checked.

Device Font

This option is for Windows only. If checked, it forces Windows to substitute fonts for fonts within a Flash movie that the user doesn't have.

Quality

The Quality settings affect how the Flash movie is displayed on playback. Quality is directly related to the processor speed of users' system. You have to decide if the quality of the images (vectors and bitmaps) takes precedence over the speed of the animation or vice versa.

Low

Selecting Low disables anti-aliasing, making for jagged images in your Flash movies. It does, however, ensure the best playback speed.

Auto Low

Selecting Auto Low causes Flash to enable anti-aliasing if the user's processor speed can maintain the frame rate with it enabled. Otherwise, anti-aliasing is disabled.

Auto High

Selecting Auto High causes Flash to disable anti-aliasing if the user's processor speed cannot maintain the specified frame rate.

High

Selecting this setting enables anti-aliasing at all times, even if the user's processor cannot maintain the specified frame rate. High limits its anti-aliasing to vectors during animation, it does not smooth animated bitmaps. This is the default Quality setting.

Best

Selecting this setting enables anti-aliasing at all times for vectors and bitmaps, animated or otherwise, even if the user's processor speed cannot maintain the specified frame rate.

Window Mode

This setting is a Windows and Explorer only setting.

Window

The default setting. This places the Flash movie in its own Window within the web page for the best playback performance.

Opaque Windowless

Select this option to enable objects placed beneath the Flash movie (in CSS Layers) do not show through the Flash movie.

Transparent Windowless

Selecting this option will enable the background and any content placed behind a Flash movie in HTML show through any transparent content in the Flash movie.

Alignment

Select from Default (center), Left, Right, Top, Bottom to specify the alignment of the Flash movie in the HTML document.

Scale

Scale only has an effect if you specified dimensions other than those of the movie.

Default (Show All)

Displays the entire movie within the specified dimensions and maintains the aspect ratio. Borders may appear if the aspect ratio of the dimensions you've input differs from that of the movie.

No Border

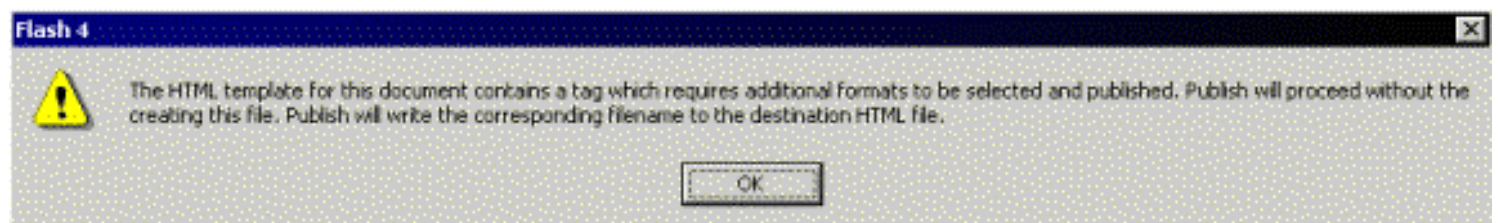
Displays the entire movie within the specified dimensions and maintains the aspect ratio. Borders will not appear if the aspect ratio of the dimensions you've input differs from that of the movie. However, clipping or cropping of content may occur as a result.

Exact Fit

Scales the movie to the exact dimensions you've specified and does not maintain the aspect ratio of the movie, which may cause your content distortion.

Show Warning Messages

If you select an HTML template that requires you to set additional settings, such as including a JPEG for an image map if the user doesn't have Flash, a warning message results when you attempt to publish the movie and the HTML file.



We selected the Image Map template, which requires us to check either GIF under Formats so that Flash can generate an image map based on our Flash content. Because we didn't enable either option, Flash warns us that we haven't done everything required to generate the proper published file.

Exporting Content in other Media

If you wish, you can export single frames or entire animations in formats other than Flash by selecting the appropriate format in the Formats section of Publish Settings. Select from GIF, JPEG, PNG, and Quicktime. Each format has particular settings which you can set by selecting the appropriate tab.

Alternately, you can select File/Export Movie, or File/Export Image to export particular frames or the entire animation in a variety of formats. You could use Flash to create an animated GIF for example.

Windows Projector/Macintosh Projector

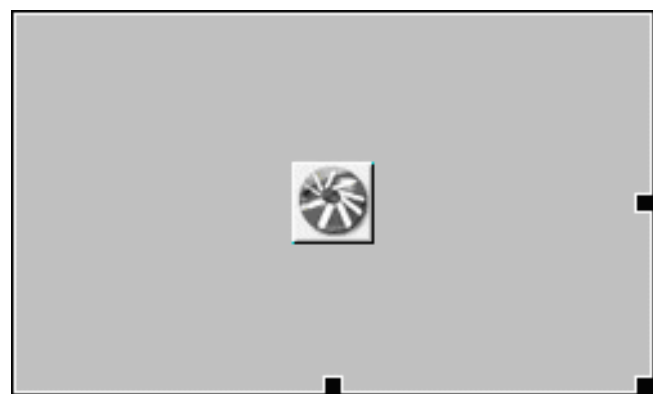
Flash can generate an executable file to play your Flash movies outside of a browser and on a system that does not have Flash installed. This is useful if you develop with Flash for projects outside the web, such as dynamic presentations for example.

Using Dreamweaver

If you use Macromedia's Dreamweaver, there is direct support for Flash. In the Dreamweaver Document Window, simply place your cursor where you wish to place

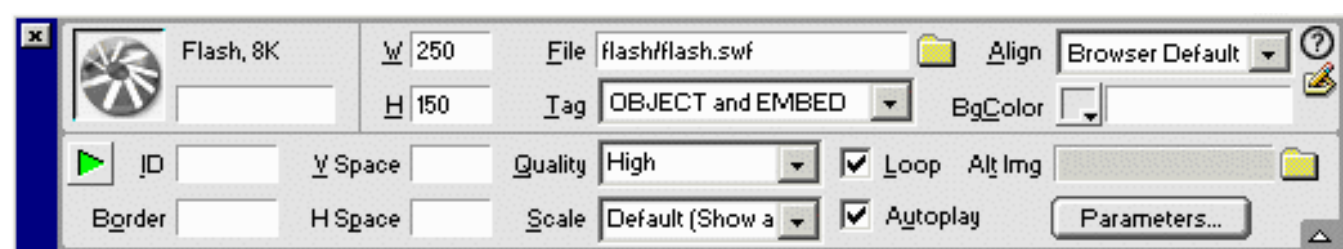
the Flash movie and {Click} the Flash icon  on the Common Objects Palette.

Navigate within your site to the folder you've placed your Flash file(s) in and {Double-Click} the .swf file. In Dreamweaver, a box appears with the Flash logo in the center. The box is the size of the movie.



The Flash movie in Dreamweaver's Document Window.

When you select the Flash movie in the Document Window, the Properties Inspector permits you to set the parameters for the Flash movie in the same manner as in the Flash Publish Settings.



The Properties Inspector for Flash files in Dreamweaver.

Dreamweaver lets you preview Flash movies by {Clicking} on the Play  and Stop  buttons in the Properties Inspector. Set settings such as Quality, Scale, Loop, etc. just as you do with the Flash Publish Settings.

Exercise

1. Open the file you created early titled MyForm fla.
2. Use Flash's Publish settings to publish your Flash movie in an HTML file.
3. Preview the file in a browser and check that your Form works, and that the button to your favorite web site is functional.
4. *{Click}* on the email button you created. It should launch your email program.

Review Questions

1. What purpose does a looping Movie Clip serve with Preloaders?
2. There are two approaches to building Preloaders. What are they?
3. Why do we use a separate Scene for Preloaders?
4. How do you customize the publish settings in Flash?
5. What does Protect from Import prevent other Flash users from doing?
6. How does the Quality setting affect the display and playback of your movies?

Summary

As a result of this chapter, you should be able to

- Create a Preloader
- Use Flash to publish your movie to an HTML page
- Use Dreamweaver to place Flash movies in HTML page

