

MX



macromedia®

FLASH™MX

2004

Learning Flash


Trademarks

Add Life to the Web, Afterburner, Aftershock, Andromedia, Allaire, Animation PowerPack, Aria, Attain, Authorware, Authorware Star, Backstage, Bright Tiger, Clustercats, ColdFusion, Contribute, Design In Motion, Director, Dream Templates, Dreamweaver, Drumbeat 2000, EDJE, EJIPT, Extreme 3D, Fireworks, Flash, Flash Lite, Flex, Fontographer, FreeHand, Generator, HomeSite, JFusion, JRun, Kawa, Know Your Site, Knowledge Objects, Knowledge Stream, Knowledge Track, LikeMinds, Lingo, Live Effects, MacRecorder Logo and Design, Macromedia, Macromedia Action!, Macromedia Breeze, Macromedia Flash, Macromedia M Logo and Design, Macromedia Spectra, Macromedia xRes Logo and Design, MacroModel, Made with Macromedia, Made with Macromedia Logo and Design, MAGIC Logo and Design, Mediamaker, Movie Critic, Open Sesame!, Roundtrip, Roundtrip HTML, Shockwave, Sitespring, SoundEdit, Titlemaker, UltraDev, Web Design 101, what the web can be, and Xtra are either registered trademarks or trademarks of Macromedia, Inc. and may be registered in the United States or in other jurisdictions including internationally. Other product names, logos, designs, titles, words, or phrases mentioned within this publication may be trademarks, service marks, or trade names of Macromedia, Inc. or other entities and may be registered in certain jurisdictions including internationally.

Third-Party Information

This guide contains links to third-party websites that are not under the control of Macromedia, and Macromedia is not responsible for the content on any linked site. If you access a third-party website mentioned in this guide, then you do so at your own risk. Macromedia provides these links only as a convenience, and the inclusion of the link does not imply that Macromedia endorses or accepts any responsibility for the content on those third-party sites.

Speech compression and decompression technology licensed from Nellymoser, Inc. (www.nellymoser.com).

 Sorenson™ Spark™ video compression and decompression technology licensed from Sorenson Media, Inc.

Opera® browser Copyright © 1995-2002 Opera Software ASA and its suppliers. All rights reserved.

Apple Disclaimer

APPLE COMPUTER, INC. MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING THE ENCLOSED COMPUTER SOFTWARE PACKAGE, ITS MERCHANTABILITY OR ITS FITNESS FOR ANY PARTICULAR PURPOSE. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED BY SOME STATES. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY PROVIDES YOU WITH SPECIFIC LEGAL RIGHTS. THERE MAY BE OTHER RIGHTS THAT YOU MAY HAVE WHICH VARY FROM STATE TO STATE.

Copyright © 2004 Macromedia, Inc. All rights reserved. This manual may not be copied, photocopied, reproduced, translated, or converted to any electronic or machine-readable form in whole or in part without prior written approval of Macromedia, Inc.

Acknowledgments

Director: Erick Vera

Project Management: Julee Burdekin, Erick Vera

Writing: Jay Armstrong, Jody Bleyle, Mary Burger, Francis Cheng, Jen deHaan, Stephanie Gowin, Phillip Heinz, Shimul Rahim, Samuel R. Neff

Managing Editor: Rosana Francescato

Editing: Mary Ferguson, Mary Kraemer, Noreen Maher, Antonio Padiál, Lisa Stanziano, Anne Szabla

Production Management: Patrice O'Neill

Media Design and Production: Adam Barnett, Christopher Basmajian, Aaron Begley, John Francis

Second Edition: June 2004

Macromedia, Inc.
600 Townsend St.
San Francisco, CA 94103

CONTENTS

INTRODUCTION: Learning Macromedia Flash MX 2004	9
Additional resources for learning Flash	10
CHAPTER 1: Create a Document	13
Take a tour of the user interface.	14
Change background and Stage size	16
Change your view of the Stage.	16
View the Library panel	17
Add graphics to the Stage	17
Add video	17
View object properties	18
Add video control behaviors	18
Use the Movie Explorer to view the document structure	19
Test the document.	19
Find help.	20
Summary.	20
CHAPTER 2: Create Accessible Flash Content	21
Set up your workspace.	22
Specify that your document be accessible to screen readers	22
Provide a document title and description.	23
Provide a title and description for instances	23
Specify that screen readers ignore elements in your document	23
Change static text to dynamic text for accessibility	24
Control the tab order and reading order	24
About testing your document with screen readers	26
Summary.	26

CHAPTER 3: Write Scripts with ActionScript	27
Set up your workspace	28
Create an instance of a symbol	28
Name button instances	29
Initialize the document	30
About ActionScript syntax	30
Locate ActionScript reference documentation	31
Add comments to ActionScript	31
Write a function for a button	31
Copy and modify a button function	32
Check syntax and test your application	33
Summary	33
CHAPTER 4: Create an Application	35
Set up your workspace	35
Copy input and dynamic text fields	36
Name text fields	36
Add and name a Button component	36
Declare variables and values for the prices	37
Specify values for input text fields	37
Write a function	38
Write an event handler for the component	39
Test your application	40
Summary	40
CHAPTER 5: Work with Layers	41
Set up your workspace	42
Select a layer	43
Hide and show layers	43
Lock a layer	43
Add and name a layer	44
Change the order of layers	44
Organize layers in a folder	44
Add a mask layer	45
Add a guide layer	45
Delete a layer	46
Summary	46
CHAPTER 6: Create a User Interface with Layout Tools	47
Set up your workspace	48
Use guides to align an object	48
Change the Stage size	49
Resize objects to match the Stage size	49
Specify snap alignment settings	50
Align an object using the alignment guides	50
Align objects using the Align panel	51
Snap objects to each other	51

Align objects using the Property inspector	52
Align objects using the grid and arrow keys	53
Summary	53
CHAPTER 7: Draw in Flash	55
Set up your workspace	56
Select a shape tool	56
Select options to create a polygon	57
Draw a polygon	57
Rotate the shape	57
Use the cut-out feature	58
Transform the shape of your drawing	58
Copy strokes	59
Draw with the Line tool	59
Select and add a different fill color	59
Group the shape	60
Create a logo with the Pen tool	60
Summary	61
CHAPTER 8: Create Symbols and Instances	63
Set up your workspace	63
About creating symbols	64
Create a graphic symbol	64
Duplicate and modify an instance of a symbol	65
Modify a symbol	65
Create a movie clip symbol	66
Assign an instance name to the movie clip	66
Add an effect to the movie clip	67
Summary	67
CHAPTER 9: Add Animation and Navigation to Buttons	69
Set up your workspace	70
Create a button from grouped objects	70
Name a button instance	71
View the hit area by enabling buttons	71
Change the hit area of a button	71
Align buttons	72
Create animation for a button state	73
Add an action to a button	73
Add navigation to a button	74
Test the SWF file	74
Summary	75

CHAPTER 10: Add Static, Input, and Dynamic Text	77
Set up your workspace	78
Create an expanding-width text block	79
Create a fixed-width text block	79
Edit text and change font attributes	80
Select device fonts	80
Add an input text field	81
Copy a text field	81
Assign instance names to text fields	82
Create a dynamic text field	82
Specify format options	83
View ActionScript for the dynamic text field	83
Test the SWF file	83
Check spelling	84
Summary	84
CHAPTER 11: Create a Timeline Animation	85
Set up your workspace	86
Create a motion tween	86
Create a shape tween	87
Copy and paste keyframes in an animation	88
Change the speed of the animation	89
Test the SWF file	90
Summary	90
CHAPTER 12: Create a Presentation with Screens (Flash Professional Only)	91
Set up your workspace	91
View the screen hierarchy and screen Timelines	92
View screen properties	93
Add content to a presentation slide	93
Add screen navigation behaviors to buttons	94
Add and name a slide	94
Select and move slides	95
Add content to a new slide	95
Add transition behaviors	95
Test your presentation	96
Summary	96

CHAPTER 13: Add Interactivity with ActionScript	97
Set up your workspace.	98
Name button instances	99
Add a scene	99
Move between scenes.	99
Control the document with a stop() action	100
Link a button to a scene	100
Add navigation to return to Scene 1	100
Play a movie clip	101
Use a behavior to play an MP3 file	103
Test your document.	103
Summary.	103
CHAPTER 14: Create a Form with Conditional Logic and Send Data	105
Set up your workspace.	105
Add an input text field to collect form data	106
Add a Submit button to the form	107
Add an error message.	107
Add a confirmation message	108
Add a stop() action	108
Add frame labels for navigation.	109
Add conditional logic for the Submit button.	109
Pass data out of a SWF file	110
Write a function for the Try Again button.	110
Test your SWF file.	111
Summary.	111
CHAPTER 15: Work with Objects and Classes Using ActionScript 2.0	113
Set up your workspace.	113
About classes and object types	114
Creating an object from a class	114
Creating a custom class	115
Create two objects from the Product class	118
About extending existing classes	119
Extend the MovieClip class to create a new class	120
Summary.	121

INTRODUCTION

Learning Macromedia Flash MX 2004

Learning Flash overview

Macromedia Flash MX 2004 and Macromedia Flash MX Professional 2004 provide everything you need to create and deliver rich web content and powerful applications. Whether you're designing motion graphics or building data-driven applications, Flash has the tools necessary to produce great results and deliver the best user experience across multiple platforms and devices.

This lessons in this book are designed to introduce you to Flash. As you complete the lessons, you will learn many of the basic tasks of creating Flash applications.

Note: This book is not a comprehensive manual detailing all the features of Macromedia Flash MX 2004. For in-depth information about using Flash, from within the Flash application, select Flash Help (Help > Help).

About the lessons

The book includes several step-by-step lessons, designed to teach you the fundamentals of Flash. We recommend that you go through the lessons using the sample files provided. The path to the sample file is provided in each lesson.

By completing these hands-on lessons, you'll learn how to use Flash to add text, graphics, and animation to your Flash applications. Additionally, you'll learn how easy it is to customize your Flash application by using ActionScript and behaviors.

The lessons are targeted toward beginners to intermediate-level Flash designers and developers who want to get up to speed quickly.

Each lesson focuses on a specific Flash design feature or topic and takes approximately 10–20 minutes to complete, depending on your experience. You can follow the lessons in this book in order, or you can start with a lesson or task that best suits your interests and experience.

The Quick Start, Basic Flash, and Basic ActionScript lessons provide an opportunity to explore the Flash workspace. In these lessons, you learn how to create a Flash document, write ActionScript, work with video and video control behaviors, and add a Flash component.

Where to start

If you are new to Flash, start with the “Getting Started with Flash” chapter in Help (Help > Help > Using Flash > Getting Started with Flash) to help you become familiar with Flash before proceeding to the Flash lessons.

To complete the Flash lessons, you can follow the lessons in this book or open the lessons in the How Do I panel in Flash Help (Help > How Do I). Although the lessons can be completed in any order, the best way to build your understanding of Flash is to take the lessons in the order they appear in the How Do I panel.

Typographical conventions

The following typographical conventions are used in this book:

- Menu items are shown in this format: menu name > menu item name. Items in submenus are shown in this format: menu name > submenu name > menu item name.
- Code font indicates HTML tag and attribute names as well as literal text used in examples.
- *Italic code font* indicates replaceable items (sometimes called *metasymbols*) in code.
- **Bold roman text** indicates text that you should enter verbatim.

Additional resources for learning Flash

Flash contains a variety of media to help you learn the program quickly and become proficient in creating your own Flash applications.

About the electronic manuals and lessons

The following electronic manuals are available through the Help panel (Help > Help):

- *Getting Started with Flash* (or Getting Started Help) provides a hands-on introduction to Flash.
- *Using Flash* (or Using Flash Help) contains complete information on working in the Flash authoring tool, including all commands, features, and user interface elements.
- *Using Components* contains information on adding and configuring components in a Flash document, as well as information on creating components.
- *Using ActionScript in Flash* (or Using ActionScript Help) provides a conceptual introduction to the ActionScript language
- *Flash ActionScript Language Reference* (or Flash ActionScript Language Reference Help) documents all actions, methods, and properties in the ActionScript API.

If you are a new user, you'll find the following information useful:

- *Getting Started with Flash* (or Getting Started Help) provides basic information about the Flash workspace and guides you in creating a simple document.
- The lessons on the How Do I tab offer an introduction to using Flash. Refer to the Quick Start, Basic Flash, and Basic ActionScript lessons (Help > How Do I).

Using the Help panel

The updatable Help panel provides access to information on how to use Flash. For more information about the Help panel, see the following sections.

Accessing help

The tabs in the Help panel—Help and How Do I—contain the full set of user assistance information provided with the Flash application.

- You can select the Help tab to display general help information.
- You can select the How Do I tab to display a list of 10- to 20-minute lessons that guide you through many of the features in Flash.

To access help and the table of contents:



1. Select Help > Help to open the Help panel.
2. If the table of contents is not visible, click the Table of Contents button to display it.
A list of help books is displayed. The Help tab is selected by default.
3. Click a book to open it and display its topics.
4. Click a topic to select it.

The How Do I tab contains brief lessons that introduce you to the main features of Flash, letting you practice on isolated examples. If you are new to Flash, or if you have used only a limited set of features, start with the How Do I tab.

To start the lessons:

1. Select Help > How Do I to display the How Do I tab of the Help panel.
In the table of contents, you'll find a list of lessons.
2. Click a lesson topic to open it.

To search for a word or phrase in the Help panel:

1. Click the Search icon in the Help panel toolbar.
2. Type a word or phrase in the text box, then click Search.
A list of topics that contain the word or phrase is displayed.

To close the help table of contents:

- Click the X above the table of contents.

Note: You can access the table of contents again by clicking the Table of Contents button.

To print a help page:



1. Click the Print button in the Help panel toolbar.
2. In the Print dialog box, select the printer and other printing options, then click Print.

Updating the Help panel

The Update feature allows you to update your help system with new and revised documentation, including procedures and lessons. You can click the Update button to see if new information is available. Additionally, if you see a topic in help with the text “For the latest information about this topic, click the Update button at the top of the Help tab,” you can click this button to update Flash Help.

To update Flash Help:

1. Verify that you’re connected to the Internet.
2. Click the Update button in the Help panel toolbar and follow instructions to download the help system.



CHAPTER 1

Create a Document

You're about to experience the power of Macromedia Flash MX 2004 and Macromedia Flash MX Professional 2004. You'll see how, in a few minutes, you can create a compelling web experience that combines video, text, graphics, and media control behaviors. In the process, you'll learn about the Flash authoring environment as you accomplish the following tasks:

- Tour the user interface
- Dock and undock panels
- Change the background and Stage size
- Change your view of the Stage
- View your document library
- Add graphics to the Stage
- Add video
- View object properties
- Add video control behaviors
- Use the Movie Explorer to view the document structure
- Test the document
- Find help

Before taking this lesson, we recommend that you read *Getting Started with Flash*, to learn about the Flash workspace. To access this guide, select Help > Getting Started in Flash Help.

Take a tour of the user interface

First you'll open the starting FLA file that you'll use to complete this lesson. Each lesson includes one start file, and a finished file that demonstrates how the FLA file should appear upon completion of the lessons.

1. To open your start file, in Flash select File > Open and navigate to the file:
 - On Windows 2000 or XP, browse to *boot drive*\ Documents and Settings\All Users\Application Data\Macromedia\Flex MX 2004*language*\Configuration\ HelpPanel\ HowDoI\ QuickTasks\start_files and double-click document_start fla.
 - On Windows 98, browse to *boot drive*\Windows\Application Data\Macromedia\Flex MX 2004*language*\Configuration\HelpPanel\ HowDoI\QuickTasks\start_files and double-click document_start fla.
 - On Macintosh, browse to *Macintosh HD*/Users/Shared/Application Support/Macromedia/Flex MX 2004/*language*/Configuration/HelpPanel/HowDoI/QuickTasks/start_files and double-click document_start fla.

Note: The QuickTasks\finished_files folder contains completed versions of lesson FLA files, for your reference.

The document opens in the Flash authoring environment. The document already includes two layers in the Timeline. To learn more about layers, select Help > How Do I > Basic Flash > Work with Layers.

One of the layers is a guide layer, which assists you in placing objects on the Stage. The other layer is a content layer in which to place the objects that will compose your document.

Note: Guide layers do not appear in the Flash content that you test or publish.

2. Select File > Save As and save the document with a new name, in the same folder, to preserve the original start file.
As you complete this lesson, remember to save your work frequently.

Select panel sets and arrange panels

The Training Layout panel set arranges your workspace in a way that facilitates taking lessons. You'll use this layout for all lessons that you take in Flash.

- Select Window > Panel Sets > Training Layout.
You can move panels around, and resize them, as follows:
 - You can undock a panel by clicking the upper left corner of the panel, in the Title bar, and dragging the panel to another location in the work area.
If the panel snaps against a border, it is docked in a new location (or docked in the same location, if you moved it back). Otherwise, the panel is undocked.
 - You can resize an undocked panel by dragging the lower right edge out to enlarge the panel.

Use tools to create Flash content

The white rectangular Stage area is where you can arrange objects as you wish them to appear in your published file.

Note: In Windows operating systems, you can open several documents at once and use document tabs, above the Stage, to navigate between them.

The Tools panel, next to the Stage, offers a variety of controls that let you create text and vector art. To learn more about Tools panel tools, select Help > How Do I > Basic Flash > Draw in Flash and Help > How Do I > Basic Flash > Add Static, Input, and Dynamic Text.



1. Click the Pencil tool in the Tools panel. Click the Stroke color box, in the Tools panel Colors area, and select any color except white.
2. Click and drag around the Stage, without releasing the mouse, to draw a line.

You've created Flash content. Your finished document will be much more impressive.

Undo actions

Flash can undo a series of changes to your document. You'll undo the artwork that you just created.

1. To see the undo feature in action, first open the History panel (Window > Other Panels > History).

The Pencil tool appears in the panel, because using the tool was your last action.

2. Do one of the following:
 - Select Edit > Undo Pencil Tool.
 - Press Control+Z (Windows) or Command+Z (Macintosh).

Your scribbles disappear from the Stage. The History panel now shows a dimmed Pencil tool, which indicates the undo action was executed.

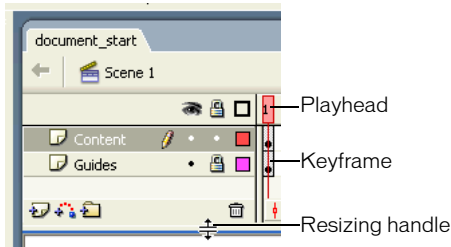
Flash, by default, is set to undo 100 of your changes, in reverse order of execution. You can change the default setting in Preferences. To change your preferences, see Setting Preferences in Flash in the "Getting to Know the Workspace" chapter of *Getting Started with Flash* (Help > Getting Started).

3. To close the History panel, click the options menu in the upper right corner of the panel and select Close Panel.

View the Timeline

Just above the Stage, you see the Timeline and layers. You can create and name layers, then add content to frames on layers to organize how your Flash content plays as the playhead moves across the frames.

- Move the mouse pointer over the area that separates the Stage from the Timeline. When the resizing handle appears, drag up or down slightly to resize the Timeline as necessary.



The playhead (the red indicator line) is on Frame 1 in the Timeline. The keyframes are designated by small circles in the frames, which are filled, indicating there's content in those frames. You can add a keyframe to a document when you want the Flash content to change in some way in that frame.

Change background and Stage size

The Stage provides a preview of how your Flash content will appear in your published file. You'll change the size of the Stage, to accommodate artwork designed for a larger Stage, and you'll change the background color of the Stage.



1. In the Tools panel, click the Selection tool.
2. On the Stage, click anywhere in the gray work area that surrounds the Stage, or on the background area of the Stage, so that no objects are selected.

The Property inspector, below the Stage, displays properties for the document when no objects are selected.

3. To change the Stage background color, click the Background color box and select a light shade of gray, such as gray with the hexadecimal value of #CCCCCC.
4. To change the Stage size, click the Size button in the Property inspector. In the Document Properties dialog box, enter **750** for the Stage width, then click OK.

The Stage resizes to 750 pixels wide.

Change your view of the Stage

You can change your view of the Stage without affecting the actual Stage size of your document.

1. In the Stage View text box, above the right side of the Stage, type **500%**. Then press Enter or Return.

Your view of the Stage enlarges to 500%.

2. In the Stage View pop-up menu, which you access by clicking the control to the right of the text box, select **100%** to view the Stage in dimensions that correspond to the size of the published Flash content.

View the Library panel

Flash content that you import or that is a symbol is stored in your Library panel. To learn more about symbols and instances, select Help > How Do I > Basic Flash > Create Symbols and Instances.

- To view the Library panel, select Window > Library.

We've already imported library items and created symbols for objects that you'll use in this lesson.

Note: Flash also contains a library of buttons that you can use in your document. To view this library, after taking this lesson, select Window > Other Panels > Common Libraries and select the Buttons library.

Add graphics to the Stage

To add library items to your document, you verify that you're adding the object to the correct layer, then drag the item from the Library panel to the Stage.

1. In the Timeline, click the Content layer name to select that layer. With the Selection tool selected, drag the Title movie clip, which contains a bitmap image and vector graphic, from the Library panel to the Stage and align it over the Title guide.

In Flash, you can work with both bitmap images, which describe graphics using pixels, and vector art, which uses mathematical representation to describe art. For more information, see "About vector and bitmap graphics" in *Using Flash*.

2. With the Content layer still selected, drag the text symbol from the Library panel to Stage, and align it with the Trio ZX2004 text that's already in place as a guide. You can use your keyboard arrow keys to nudge the text into place.

The title text is actually a graphic created from text.

Add video

The Library panel includes an imported Flash video file (FLV). You'll add the video to your document, and Flash will add the necessary frames to play the video.

To learn more about using video in Flash, see "Working with Video" in *Using Flash*.

1. Verify that the Content layer is still selected in the Timeline. From the Library panel, drag the `ggb_movie_for_trio_new` video to the dark gray Video guide on the Stage.
2. A dialog box appears that indicates Flash will add 138 frames to the Timeline for the video. Click Yes.
3. Drag the playhead across the Timeline to view the video.

View object properties

When you add an object to the Stage, you can select it, then view and change its properties in the Property inspector. The type of object selected determines which properties appear. For example, if you select a text object (not a text graphic, which we use in this lesson), the Property inspector displays settings such as font, type size, and paragraph formatting, which you can either view or change. If no object is selected, the Property inspector displays properties for the entire document.

1. On the Stage, with the Selection tool selected, click the Title graphic.

The Property inspector (Window > Properties) shows specifications, such as height, width, and Stage coordinates, for the grouped object.

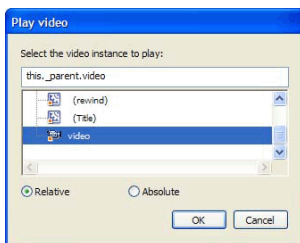
2. On the Stage, click the bounding box for the video movie clip that you dragged to the Stage and view its attributes in the Property inspector.
3. In the Instance Name text box of the Property inspector, enter **video** as the instance name.

Note: Because ActionScript, the Flash scripting language, often refers to instance names, it is a good practice to name instances. To learn more about naming instances, select Help > How Do I > Quick Tasks > Write Scripts with ActionScript.

Add video control behaviors

Behaviors let you add complex functionality to your document easily, without having to know ActionScript, the Flash scripting language. You'll now add behaviors for video control.

1. In the Timeline, click Frame 1 of the Content layer to select it, if it's not already selected.
2. On the Stage, click the Play movie clip instance (which looks like a play button) to select it. In the Behaviors panel (Window > Development Panels > Behaviors), click the Add (+) button and select Embedded Video > Play. To learn more about symbols and instances, select Help > How Do I > Basic Flash > Create Symbols and Instances.
3. In the Play Video dialog box, verify that Relative is selected. Select video, which is the instance name that you gave to the video clip, then click OK.



4. On the Stage, click the Pause movie clip instance to select it. In the Behaviors panel, click the Add (+) button and select Embedded Video > Pause.
5. In the Pause Video dialog box, again select the video movie clip, then click OK.
6. On the Stage, click the Rewind movie clip instance to select it. In the Behaviors panel, click the Add (+) button and select Embedded Video > Rewind.



7. In the Rewind Video dialog box, select video.
8. In the Number of Frames to Step Back text box, enter 20.

The Number of Frames to Step Back text box indicates how many frames the playhead should move back when the user clicks the Rewind button.

Note: Additional video control behaviors let you fast-forward, hide, and show a video.

Use the Movie Explorer to view the document structure

The Movie Explorer helps you arrange, locate, and edit media. With its hierarchical tree structure, the Movie Explorer provides information about the organization and flow of a document.

1. Select Window > Other Panels > Movie Explorer.

If necessary, enlarge the Movie Explorer to view the tree structure within the pane.

The Movie Explorer filtering buttons display or hide information.

2. Click the pop-up menu in the title bar of the Movie Explorer, and select Show Movie Elements and Show Symbol Definitions, if they're not already selected.
3. Configure the filtering buttons, along the top of the Movie Explorer, so the only ones selected are Show Buttons, Movie Clips, and Graphics; Show Action Scripts; and Show Video, Sounds, and Bitmaps.

If you move your mouse pointer over a button, a tooltip displays the name of the button.

Examine the list to view some of the assets included in the document, and to see their relationship to other assets.

4. In the Movie Explorer pane, expand Actions for Play to view ActionScript that Flash created when you added the Play video control behavior.
5. To close the Movie Explorer, click its close box.

Test the document

As you author a document, you should save and test it frequently to ensure the Flash content plays as expected. When you test the SWF file, click the video control buttons to see if the video stops, plays, and rewinds as expected.

1. Save the document (File > Save) and select Control > Test Movie.

The Flash content plays in a SWF file window. While FLA is the extension for documents in the authoring environment, SWF is the extension for tested, exported, and published Flash content.

2. When you finish viewing the SWF content, close the SWF file window to return to the authoring environment.

Find help

The lessons provide an introduction to Flash, and suggest ways that you can use features to create exactly the kind of document required. For comprehensive information about a feature, procedure, or process described in the lessons, see the Help tab of the Help panel (Help > Help).

Summary

Congratulations on creating a Flash document that includes graphics, a video, and video control behaviors. In a few minutes, you learned how to accomplish the following:

- Tour the user interface
- Dock and undock panels
- Change the background and Stage size
- Change your view of the Stage
- View your document library
- Add graphics to the Stage
- Add video
- View object properties
- Add video control behaviors
- Use the Movie Explorer to view the document structure
- Test the document
- Find help

To learn more about Flash, take another lesson.

CHAPTER 2

Create Accessible Flash Content

With knowledge of a few design techniques and accessibility features in Macromedia Flash MX 2004 and Macromedia Flash MX Professional 2004, you can create Flash content that is accessible to all users, including those with disabilities. This lesson demonstrates how to create an accessible document, designed for use with screen readers (which read web content aloud for visually impaired users) and other assistive technologies, as you accomplish the following tasks:

- Specify that your document be accessible to screen readers
- Provide a document title and description
- Provide a title and description for document instances
- Specify that screen readers ignore elements in your document
- Change static text to dynamic text for accessibility
- Control the order in which users navigate with the Tab key
- Control the reading order with ActionScript

The lesson offers an introduction to basic techniques of making your Flash content accessible. For detailed and comprehensive information about incorporating accessibility features in your Flash content, see “Creating Accessible Content” in *Using Flash*.

Set up your workspace

First, you'll open the start file for the lesson and set up your workspace to use an optimal layout for taking lessons.

1. To open your start file, in Flash select File > Open and navigate to the file:
 - On Windows 2000 or XP, browse to *boot drive*\ Documents and Settings\All Users\Application Data\Macromedia\Flex MX 2004*language*\Configuration\ HelpPanel\ HowDoI\ QuickTasks\start_files and double-click accessibility_start fla.

Note: If the Application Data folder is hidden, you'll need to change your Windows Explorer settings to see the folder.

 - On Windows 98, browse to *boot drive*\Windows\ Application Data\Macromedia\Flex MX 2004*language*\Configuration\HelpPanel\ HowDoI\QuickTasks\start_files and double-click accessibility_start fla.
 - On Macintosh, browse to *Macintosh HD*/Users/Shared/Application Support/Macromedia/ Flex MX 2004/*language*/Configuration/HelpPanel/HowDoI/QuickTasks/start_files and double-click accessibility_start fla.
2. Select File > Save As and save the document with a new name, in the same folder, to preserve the original start file.

Note: As you complete this lesson, remember to save your work frequently.
3. Select Window > Panel Sets > Training Layout to modify your workspace for taking lessons.

Specify that your document be accessible to screen readers

You'll now specify that your document be accessible to screen readers, and provide a name and description of your document that a screen reader can read aloud.

1. With nothing selected on the Stage, select Window > Other Panels > Accessibility.
2. In the Accessibility panel, verify that the following options are selected:
 - Make Movie Accessible** is selected by default and allows Flash Player to pass accessibility information to a screen reader.
 - Make Child Objects Accessible** allows Flash Player to pass accessibility information nested inside a movie clip to a screen reader. If this option is selected for the entire document, you can still hide child objects for individual movie clips.
 - Auto Label** associates text next to another Stage object, such as an input text field, as a label or title for that element.

Provide a document title and description

In the Accessibility panel for the document, you can enter a name and description for your document for screen readers.

- In the Name text box, enter **Trio ZX2004**. In the Description text box, enter **Corporate website about the Trio ZX2004. Includes 6 navigation buttons, overview text, and an animated car.**

Provide a title and description for instances

Now that you've provided information about the entire document, you can provide information about Stage objects included in the document.

1. Select the Trio Motor Company logo along the top of the Stage. In the Accessibility panel, enter **Trio Motor Company** in the Name text box. Do not enter anything in the Description text box.

Not every instance needs a description, which is read with the title information. If the title name sufficiently describes the function of the object, then you don't need to include a description.

2. With the Accessibility panel still open, select the Dealers button on the Stage.

Information in the Accessibility panel changes to reflect Accessibility options for the selected object.

In the Accessibility panel for the Dealers button, you do not need to provide a name in the Title text box, because the button includes a text label that the screen reader will read. If you did not want the screen reader to read the text in the button, you could deselect Auto Label when you set up accessibility for the document.

3. In the Description text box, enter **Links to a web page with information about dealers nationwide.**

The other buttons also include text, which the screen reader will read aloud; therefore, you do not need to provide a title. Since the title of the buttons is fairly self-explanatory, there's no need for you to include descriptions.

Specify that screen readers ignore elements in your document

Screen readers follow a specific order when reading web content. However, when content on the web page changes, most screen readers will begin reading the web content all over again. This screen reader feature can be problematic when Flash content contains, for example, animation, which could cause the screen reader to begin again each time there's a change in the animation.

Fortunately, you can use the Accessibility panel to either deselect Make Object Accessible, so that the screen reader does not receive accessibility information about the object, or deselect Make Child Objects Accessible, so that the screen reader does not receive accessibility information nested within a movie clip. You'll now do the latter so that users will know the web page contains an animation, and the animation won't cause the screen reader to constantly refresh.

1. On the Stage, click the car, which is the `safety_mc` movie clip instance.
2. In the Accessibility panel, deselect **Make Child Objects Accessible**. In the Name text box, enter **Trio ZX2004 animation**. In the Description text box, enter **Animation that includes three views of the Trio ZX2004**.

Change static text to dynamic text for accessibility

Static text is accessible to screen readers. However, you cannot provide static text with an instance name, which is required to control the tab order and reading order. You'll change the overview text paragraph to dynamic text and specify accessibility options.

1. On the Stage, select the text that begins “The TRIO ZX2004 provides the ultimate in efficiency ...”

The Accessibility panel changes to indicate that you cannot apply accessibility features to this selection.

2. In the Property inspector, select **Dynamic Text** from the Text Type pop-up menu.

Accessibility settings now appear in the Accessibility panel.

3. In the Instance name text box, enter an instance name of `text9_txt`.

Note: To specify a tab order and reading order, which you'll do next, you must provide an instance name for all instances. The instance name must be unique in your document.

Control the tab order and reading order

You can create a tab order that determines the order in which objects receive focus when the users press the Tab key. You can also control the order in which a screen reader reads information about the object (known as the reading order). You can create both the tab and reading order using the `tabIndex` property in ActionScript (In ActionScript, the `tabIndex` property is synonymous with the reading order). If you have Flash MX Professional 2004, you can use the Accessibility panel to specify the tab order, but the tab index that you assign does not necessarily control the reading order.

To create a reading order, you must assign a tab index to every instance in ActionScript.

If you have Macromedia Flash MX Professional 2004, creating a tab order is as easy as entering a number in the Tab Index text box. You can then view the tab order directly on the Stage.

To create a tab order in this lesson, use one of the following procedures. To create a reading order along with a tab order, follow the procedure to control the tab order and reading order using ActionScript.

If you have Flash MX Professional 2004, you can follow this procedure to create a tab order using the Accessibility panel:

1. With the Accessibility panel open, select the logo_mc instance at the top of the Stage. In the Accessibility panel, enter 1 in the Tab Index text box.
2. Continue to select each instance on the Stage and enter a tab order number in the Tab Index text box, using information from the following table:

Instance name	Enter the following number in the Tab Index text box
logo_mc	1
dealers_btn	2
orders_btn	3
research_btn	4
text4_txt (the text above the Overview button that reads TRIO ZX2004)	5
overview_btn	6
powerplant_btn	7
news_btn	8
safety_mc	9
text8_txt	10
text9_txt	11
bevel_mc (the bar along the bottom of the Stage)	12

If you have Flash MX Professional 2004, follow this procedure to view a tab order:

- Select View > Show Tab Order.

The tab index number that you entered appears next to the instance on the Stage.



Note: A tab order created with ActionScript, rather than the Accessibility panel, does not appear when Show Tab Order is enabled.

Follow this procedure to control the tab order and reading order using ActionScript:

1. In the Timeline, select Frame 1 of the Actions layer.
2. In the Actions panel (Window > Development Panels > Actions), view the ActionScript that creates the tab index for each instance in the document.
3. If you're using Flash MX 2004, or if you're using Flash MX Professional 2004 and you did not create the tab index using the Accessibility panel, delete the `/*` and `*/` in the script to un-comment the script:

```
_root.logo_mc.tabIndex = 1;  
_root.dealers_btn.tabIndex = 2;  
_root.orders_btn.tabIndex = 3;  
_root.research_btn.tabIndex = 4;  
_root.text4_txt.tabIndex = 5;  
_root.overview_btn.tabIndex = 6;  
_root.powerplant_btn.tabIndex = 7;  
_root.news_btn.tabIndex = 8;  
_root.safety_mc.tabIndex = 9;  
_root.text8_txt.tabIndex = 10;  
_root.text9_txt.tabIndex = 11;  
_root.bevel_mc.tabIndex = 12;
```

About testing your document with screen readers

You already know the importance of regularly testing your Flash document as you create it, to ensure it performs as expected. Frequent testing is even more important when you design a document to work with assistive technologies such as screen readers. In addition to testing tab order in your SWF file, you should also test your tab order in various browsers; some browsers differ in how the user tabs to or out of Flash content. For information about resources to test your document with a screen reader, see “Testing accessible content” in *Using Flash*.

Summary

Congratulations on creating accessible Flash content. In a few minutes, you learned how to accomplish the following tasks:

- Specify that your document be accessible to screen readers
- Provide a document title and description
- Provide a title and description for document instances
- Specify that screen readers ignore elements in your document
- Change static text to dynamic text for accessibility
- Control the order in which users navigate with the Tab key
- Control the reading order with ActionScript

Macromedia maintains an extensive website devoted to accessibility. For more information about accessibility with Macromedia products, see the Macromedia accessibility website at www.macromedia.com/macromedia/accessibility.

CHAPTER 3

Write Scripts with ActionScript

The ActionScript language that is part of Macromedia Flash MX 2004 and Macromedia Flash MX Professional 2004 offers designers and developers a variety of benefits. With ActionScript you can control document playback in response to events such as elapsed time and loading data; add interactivity to a document in response to user actions, such as a button click; use built-in objects, such as a button object, with built-in associated methods, properties, and events; create custom classes and objects; and create more compact and efficient applications than you could create using user interface tools, all with code that you can reuse.

ActionScript is an object-oriented scripting language that offers control over how your Flash content plays. In subsequent lessons, you'll see how ActionScript has evolved into ActionScript 2.0 to comprise a core set of language elements that make it easier to develop object-oriented programs.

You'll now use ActionScript to accomplish the following tasks:

- Name instances following recommended practices
- Initialize a document
- Apply ActionScript syntax
- Locate ActionScript reference documentation
- Add comments to ActionScript
- Write a function
- Copy and modify a function
- Check syntax and test your application

Set up your workspace

First, you'll open the start file for the lesson and set up your workspace to use an optimal layout for taking lessons.

1. To open your start file, in Flash select File > Open and navigate to the file:
 - On Windows 2000 or XP, browse to *boot drive*\ Documents and Settings\All Users\Application Data\Macromedia\Flex MX 2004*language*\Configuration\ HelpPanel\ HowDoI\ QuickTasks\start_files and double-click *scripts_start fla*.

Note: If the Application Data folder is hidden, you'll need to change your Windows Explorer settings to see the folder.

 - On Windows 98, browse to *boot drive*\Windows\Application Data\Macromedia\Flex MX 2004*language*\Configuration\HelpPanel\ HowDoI\QuickTasks\start_files and double-click *scripts_start fla*.
 - On Macintosh, browse to *Macintosh HD*/Users/Shared/Application Support/Macromedia/Flex MX 2004/*language*/Configuration/HelpPanel/HowDoI/QuickTasks/start_files and double-click *scripts_start fla*.

Note: The QuickTasks\finished_files folder contains completed versions of lesson FLA files, for your reference.
2. Select File > Save As and save the document with a new name, in the same folder, to preserve the original start file.
3. Select Window > Panel Sets > Training Layout to modify your workspace for taking lessons.

Create an instance of a symbol

You'll drag an instance of an animated movie clip from the library to the Global Positioning System artwork on the Stage. You'll then follow the recommended practice of always naming instances—both to prompt code-hinting and because in your scripts you generally refer to instance names rather than symbol names. Code hints are the tooltips that prompt you with the correct ActionScript syntax.



1. In the Tools panel, click the Selection tool. Select the map layer in the Timeline, and click the padlock next to the map layer to unlock that layer.
2. To place the movie clip accurately, select View > Snapping. Select Snap Align and Snap to Objects if the commands are not already selected.
3. From the Library panel (Window > Library), drag *map_skewed* to the black background area of the Stage.

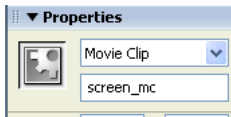
Because guides don't appear when you first drag an object from the Library panel, you'll release the object and then drag it again.

4. Drag the `map_skewed` movie clip on the Stage again so that the align guides appear. Use the guides to align the movie clip to the top and left edges of the GPS screen.



Note: If you make an error in placement, either drag the movie clip again, or press Control+Z (Windows) or Command+Z (Macintosh) to undo your changes.

5. With the instance of `map_skewed` selected on the Stage, type `screen_mc` in the Instance Name text box of the Property inspector (Window > Properties).



Flash is designed to present code hints when you name your instances with the appropriate suffix:

- When naming a movie clip instance, always give the instance a suffix of `_mc`, as in `screen_mc`.
- When naming a button, use the `_btn` suffix.
- When naming a text field, use the `_txt` suffix.

Note: For more information about additional instance name suffixes that prompt code hints, see “Writing code that triggers code hints” in *Using ActionScript in Flash*.

Name button instances

Using the appropriate suffix to prompt code hinting, you’ll provide instance names for two button instances that are already on the Stage.

1. In the Timeline, unlock the Buttons layer.
2. On the Stage, select the instance of `play_button` (the large green button).
3. In the Instance Name text box of the Property inspector, type `onButton_btn` to name the instance.
4. On the Stage, select the instance of `button_stop` (the small red button).
5. In the Instance Name text box of the Property inspector, type `offButton_btn` to name the instance.

Initialize the document

Applications have an initial state that specifies how the content first appears to users. You initialize properties and variables in the first frame of a document. You'll specify that the map movie clip not be visible when the SWF file first plays.

1. Select Frame 1 of the Actions layer. If the Actions panel isn't open, select Window > Development Panels > Actions.

Actions - Frame appears at the top of the panel, which indicates that you selected a frame in which to apply ActionScript. It's a good practice to verify that you're attaching ActionScript to the intended frame or object.

The Actions panel includes a Script pane, the blank text entry area, in which you can enter text directly; an Actions toolbox, which lets you select ActionScript to add to your script; and a Script navigator, which functions like the Movie Explorer.



2. Along the top of the Actions panel, click Insert Target path.
3. In the Insert Target Path dialog box, verify that Relative, meaning relative path, is selected. From the hierarchical tree in the dialog box, select screen_mc. Then click OK.

A target path tells ActionScript the location of an object within the overall structure of a document. For more information about paths, see "Using absolute and relative target paths" in *Using Flash*.

4. Click in the Script pane, at the end of the screen_mc text, and type a period (.).
5. When you type the period, code hints appear for the movie clip, because you used the _mc suffix when naming the instance. Double-click `_visible` from the list of code hints, and type the following:

```
= false;
```

This line of code makes the screen_mc movie clip invisible on the Stage.

Note: If code hints don't appear, you don't have code hints selected as a preference in the Actions panel. You can type `_visible` directly in the Script pane. You can also change your preferences by clicking the options menu in the upper right corner of the Actions panel. From the options menu, select Preferences, and then select Code Hints on the ActionScript tab.

Throughout authoring, remember to save your document frequently.

About ActionScript syntax

All languages, whether computer languages or written and spoken "human" languages, follow specific rules that foster comprehension. These rules are known as the language syntax.

Flash uses dot syntax, which means that the period (.) links parts of a script. Other ActionScript syntax elements include the following:

- A semicolon (;) in an ActionScript statement, like a period in an ordinary sentence, indicates the end of a statement.
- Parentheses () group arguments that apply to an ActionScript statement.
- Curly braces {} group related ActionScript statements. You can use nested braces to create a hierarchy of statements.

Later in this lesson, you'll use Flash features that allow you to test your syntax.

Locate ActionScript reference documentation

During authoring, if you'd like additional information about the ActionScript that you enter, you can select the action in the Actions toolbox or Script pane and click the Reference button. The Help panel displays information about the selected action.

1. In the Script pane of the Actions panel, double-click `visible` to select the term.

Note: After completing the next step, you'll change topics in the Help panel and you will no longer be on this lesson topic. Click the How Do I tab to return to this topic.

2. Along the top of the Actions panel, click the Reference icon.

The `visible` entry in the Help panel appears.

Add comments to ActionScript

In ActionScript, text after double slashes (`//`) is commented text, which Macromedia Flash Player ignores. Commented text often documents script functionality so that other developers can understand your script, but you can also use comments to deactivate sections of your script when debugging. As a best practice, always add comments that explain your scripts.

- In the Script pane of the Actions panel, place the insertion point at the beginning of the line of code and type `//Initialize document to hide screen movie clip`. Press Enter or Return.

Text in the Script pane appears as follows:

```
//Initialize document to hide screen movie clip.  
this.screen_mc._visible = false;
```

Note: If your commented text is many lines, you can use `/*` instead of double slashes for the beginning of the comment, and `*/` to mark the end of the comment.

Write a function for a button

A command in ActionScript is called a function. A function is a script that you can write once and use repeatedly in a document to perform a certain task. You're going to write a function that makes the `screen_mc` movie clip appear (`visible = true`) when the user releases the mouse button.

1. In the Script pane of the Actions panel, click after the last line of code, press Enter or Return twice, and type `//function to show animation`.
2. Press Enter or Return and click Insert Target Path, along the top of the Actions panel. Select `onButton_btn` from the hierarchical tree, and click OK.
3. In the Script pane, type a period (`.`) and double-click `onRelease` from the list of code hints that appears.

4. In the Script pane, press the Spacebar and type the following:

```
= function(){
```

The line of code that you just completed should appear as follows:

```
this.onButton_btn.onRelease = function(){
```

You already know how to select objects in the Insert Target Path dialog box; you'll now enter the instance names directly into the Script pane.

5. Press Enter or Return and type the following:

```
screen_mc._visible = true;
```

6. Press Enter or Return and type `};` to specify the end of the statement.

The function should appear as follows:

```
//function to show animation  
this.onButton_btn.onRelease = function(){  
    screen_mc._visible = true;  
};
```

Copy and modify a button function

You just created one function that sets the `visible` property of a movie clip to `true` when the user releases the mouse button after a button click. You can probably guess how to create another function that hides the `screen_mc` movie clip: by setting the movie clip `_visible` property to `false` when the user clicks an Off button. You'll create that function now.

1. In the Script pane, select the entire function that you just typed, including the comment, curly brackets, and semicolon. Copy the text as you normally would, using `Control+C` (Windows) or `Command+C` (Macintosh).

2. In the Script pane, place the insertion point after the last line of code. Then press Enter or Return twice, and paste the text as you normally would, using `Control+V` (Windows) or `Command+V` (Macintosh).

3. In the copied function, change the text in `onButton_btn` to read `offButton_btn`.

Remember, earlier you assigned an instance name of `offButton_btn` to an instance.

4. In the copied function, change the `visible` property of the `screen_mc` movie clip from `true` to `false`.

5. In the copied function, change the commented text after the slashes to read `function to hide animation`.

Your entire script should appear as follows:

```
//Initialize document to hide screen movie clip.  
this.screen_mc._visible = false;  
  
//function to show animation  
this.onButton_btn.onRelease = function(){  
    screen_mc._visible = true;  
};
```



```
//function to hide animation
this.offButton_btn.onRelease = function(){
    screen_mc._visible = false;
};
```

Check syntax and test your application

As you learned earlier in this lesson, ActionScript depends on correct syntax to execute properly. Flash offers a variety of ways for you to test your syntax.

1. To check the syntax, do one of the following:

- Click the options menu in the upper right corner of the Actions panel title bar and select Check Syntax.



- Click the Check Syntax button along the top of the Actions panel.

If the syntax is correct, a message appears stating that the script contains no errors.

If the syntax is incorrect, a message appears stating the script contains errors; the Output panel opens and displays information about the error.

2. Click OK to close the syntax message.

3. After you've verified that your ActionScript does not contain syntax errors, save the document and select Control > Test Movie.

When the SWF file appears, the animation should not appear in the Global Positioning System screen, because its initial `visible` property is set to `false`. When you click and release the top green button, you call the function that sets the movie clip's `visible` property to `true`. Does the animation play then? Finally, click the red Off button to see if the `visible` property for the animation is again `false`.

You can test your SWF content throughout authoring to confirm that it plays as expected.

Summary

Congratulations on learning a few basics about writing scripts with ActionScript. In a short amount of time, you learned how to accomplish the following tasks:

- Name instances following recommended practices
- Initialize a document
- Apply ActionScript syntax
- Locate ActionScript reference documentation
- Add comments to ActionScript
- Write a function
- Copy and modify a function
- Check syntax and test your application

To learn more about ActionScript, select Help > How Do I > Quick Tasks > Create an Application.

CHAPTER 4

Create an Application

The application that you'll create in this lesson lets users view the cost of selecting multiple products. A Calculate button then adds the total cost. You'll learn how to accomplish the following tasks as you create the application:

- Copy input and dynamic text fields
- Assign instance names to text fields
- Add a Button component
- Declare variables and values
- Specify values for text fields
- Write a function
- Write an event handler for the component

Set up your workspace

First, you'll open the start file for the lesson and set up your workspace to use an optimal layout for taking lessons.

1. To open your start file, in Flash select File > Open and navigate to the file:
 - On Windows 2000 or XP, browse to *boot drive*\Documents and Settings\All Users\Application Data\Macromedia\Flex MX 2004*language*\Configuration\HelpPanel\HowDoI\QuickTasks\start_files and double-click calculator_start fla.
 - On Windows 98, browse to *boot drive*\Windows\Application Data\Macromedia\Flex MX 2004*language*\Configuration\HelpPanel\HowDoI\QuickTasks\start_files and double-click calculator_start fla.
 - On Macintosh, browse to *Macintosh HD*/Users/Shared/Application Support/Macromedia/Flex MX 2004/*language*/Configuration/HelpPanel/HowDoI/QuickTasks/start_files and double-click calculator_start fla.

Note: The QuickTasks\finished_files folder contains completed versions of lesson FLA files, for your reference.

2. Select File > Save As and save the document with a new name, in the same folder, to preserve the original start file.

Note: As you complete this lesson, remember to save your work frequently.

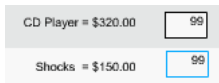
3. Select Window > Panel Sets > Training Layout to modify your workspace for taking lessons.

The form already includes an input text field in the QTY column and a dynamic text field in the Price column. You'll copy the text fields for the Shocks and Cover rows.

Copy input and dynamic text fields

You'll use input text fields to create a form.

1. Click the input text field where users enter the quantity of CD players. Press Alt and drag the copy of the field down to the Shocks QTY area.



2. Alt-click the input text field that you just dragged; then drag the new copy of the field to the Cover QTY area.
3. Alt-click the dynamic text field for the CD players price; then drag the copy of the field to the Shocks price area.
4. Alt-click the field that you just dragged; then drag the copy to the Cover price area.

Name text fields

Before you can specify values for the text fields in ActionScript, you first need to give each text field an instance name in the Property inspector. By appending the instance name with TXT, you're identifying the object as a text object.

1. Click the top input text field in the QTY column. In the Instance Name text box of the Property inspector (Window > Properties), type **qty1_txt**.
2. Follow the previous procedure to name the middle and bottom input text fields **qty2_txt** and **qty3_txt**, respectively.
3. Click the top dynamic text field in the Price column. In the Instance Name text box of the Property inspector, type **price1_txt**.
4. Follow the previous procedure to name the middle and bottom text fields in the column **price2_txt** and **price3_txt**, respectively.

Add and name a Button component

Components are movie clips that offer an easy way of adding advanced functionality to your document without having to know advanced ActionScript. You'll use the Button component to create a Calculate button that totals prices. Because the component you're using is based on ActionScript 2.0, you first need to configure your Publish Settings dialog box to ensure your Flash content plays as expected.

1. Select File > Publish Settings.
2. On the Flash tab of the Publish Settings dialog box, select ActionScript 2.0 in the ActionScript Version pop-up menu, if it's not already selected.
3. In the Timeline, click the Components layer to select it.
4. From the Components panel (Window > Development Panels > Components), drag the Button component to the Stage and place it over the Calculate guide.
5. On the Parameters tab of the Property inspector, with the Button component selected, click the Button text on the Label row, and type **Calculate**. Then press Enter or Return.
The text that you type in the Label text box is the text that appears on the component.
6. In the Instance Name text box, enter **calculate** to provide the button with an instance name.

Declare variables and values for the prices

For your application to multiply the quantity of parts selected by the price of the part, you need to define a variable for each part in ActionScript. The value for the variable is the cost of the part.

1. In the Timeline, click Frame 1 of the Actions layer and open the Actions panel (Window > Development Panels > Actions).
2. In the Script pane, type `//declare variables and values for car part prices`.

The parallel slashes (`//`) indicate that the text that follows is a comment. As a best practice, always add comments that offer an explanation of your ActionScript.

Note: As you take this lesson, you might find that you'd like to turn off code hints—the tooltips that prompt you with the correct ActionScript syntax. If so, you can turn off code hinting by clicking the options menu in the upper right corner of the Actions panel. Select Preferences, and then deselect Code Hints on the ActionScript tab.

3. Press Enter or Return and type the following to indicate the cost of each part:

```
var priceCD = 320;  
var priceShocks = 150;  
var priceCover = 125;
```

Specify values for input text fields

You'll specify values for the input text fields. You'll use the values when you write ActionScript that multiplies the quantity and cost values.

1. In the Script pane, with the insertion point after the text that reads `125;`, press Enter or Return twice and type the following comment:

```
//set initial values for the quantity text fields
```

2. Press Enter or Return, and type the following:

```
qty1_txt.text = 0;
```

`qty1_txt` is the instance name that you gave the first input text field under the QTY column. `.text` is a property that defines the initial value in the text field, which you specify is 0.

3. Press Enter or Return and type the following two lines to set values of 0 for the other two QTY fields:

```
qty2_txt.text = 0;
qty3_txt.text = 0;
```

When you finish, the ActionScript should appear as follows:

```
//set initial values for the quantity text fields
qty1_txt.text = 0;
qty2_txt.text = 0;
qty3_txt.text = 0;
```

Write a function

A function is a script that you can use repeatedly to perform a specific task. You can pass parameters to a function, and it can return a value. In this lesson, every time your user clicks the Calculate button, a function will run that multiplies data in the input text fields and returns values in the dynamic text fields. You'll write that function now.

1. In the Script pane, with the insertion point after the ActionScript that reads `qty3_txt.text = 0;`, press Enter or Return twice and type the following comment:

```
//calculate quantity times price
```

2. Press Enter or Return and type the following to create a function that runs when the playhead enters Frame 1, where you're attaching the script:

```
this.onEnterFrame = function (){
```

3. Type the following ActionScript to specify how the function should multiply the values in the input text fields for the airplane:

```
price1_txt.text = Number (qty1_txt.text)*Number (priceCD);
```

`price1_txt` is the instance name that you gave to the top price input text field on the Stage.

`.text` defines the text that should appear in the text field, which is the number of parts multiplied by the cost of the part: the \$320 that you set as the value for the `priceCD` variable.

4. Press Enter or return and type the following two lines:

```
price2_txt.text = Number (qty2_txt.text)*Number (priceShocks);
price3_txt.text = Number (qty3_txt.text)*Number (priceCover);
};
```

Your function should appear as follows:

```
//calculate quantity times price
this.onEnterFrame = function (){
    price1_txt.text = Number (qty1_txt.text)*Number (priceCD);
    price2_txt.text = Number (qty2_txt.text)*Number (priceShocks);
    price3_txt.text = Number (qty3_txt.text)*Number (priceCover);
};
```

Write an event handler for the component

For your SWF file to react to events such as a mouse click, you can use event handlers—ActionScript associated with a particular object and event. You’ll use an `on()` event handler for the Button component that calculates the total price when users click the button.

For more information about event handlers, see “Handling Events” in *Using ActionScript in Flash*.

1. On the Stage, click the Button component and go to the Actions panel.

The tab at the bottom of the Actions panel, labeled calculate, indicates that you’re attaching the script directly to the selected object rather than to a frame.

2. In the Script pane, type the following comment:

```
//Calculates total price
```

3. After the comment, press Return or Enter and type the following to create a handler for the PushButton component that you placed on the Stage:

```
on(click) {
```

You just typed the start of the `on()` event handler. The `(click)` specifies that the event should occur when the user clicks the Calculate button.

A Button component has its own Timeline. In the Timeline hierarchy, the component Timeline is a child of the main Timeline. To point to elements from the Button component Timeline to the main Timeline in this script, you use the code `with (_parent)`.

4. With the insertion point at the end of the line you just typed, press Enter or Return and type the following:

```
with(_parent){
```

5. Press Enter or Return and complete your handler by typing the following:

```
priceTotal_txt.text = Number (price1_txt.text) + Number (price2_txt.text) +  
    Number (price3_txt.text);  
    }  
}
```

When you finish, your script should appear as follows:

```
on(click) {  
    with(_parent){  
        priceTotal_txt.text = Number (price1_txt.text) + Number (price2_txt.text)  
        + Number (price3_txt.text);  
    }  
}
```

The event handler that you typed specifies that the text in the `priceTotal_txt` field should be the sum of the values in the `Price1_txt`, `Price2_txt`, and `Price3_txt` fields.

Test your application

You'll test your application to ensure that it executes as expected.

1. Save your document and select Control > Test Movie.
2. In the test version of your movie that appears in Flash Player, type numbers in the QTY fields to see what appears in the Price fields.
3. Click the Calculate button to see the total price for all parts.

Summary

Congratulations on learning how to create an application. In a few minutes, you learned how to complete the following tasks:

- Copy input and dynamic text fields
- Assign instance names to text fields
- Add a Button component
- Declare variables and values
- Specify values for text fields
- Write a function
- Write an event handler for the component

To learn more about ActionScript, take a lesson in the Basic ActionScript series on the How Do I tab of the Help panel.

CHAPTER 5

Work with Layers

In Macromedia Flash MX 2004 and Macromedia Flash MX Professional 2004, layers are analogous to transparent sheets of acetate stacked on top of each other. In the areas of a layer that don't contain content, you can see through to content on the layers below. Layers assist you in organizing content in your document. For example, you can keep background art on one layer and navigational buttons on another. Additionally, you can create and edit objects on one layer without affecting objects on another layer. In this lesson, you'll learn about layers as you accomplish the following tasks:

- Select a layer
- Hide and show layers
- Lock a layer
- Add and name a layer
- Change the order of layers
- Organize layers in a folder
- Add a mask layer
- Add a guide layer
- Delete a layer

Set up your workspace

First, you'll open the start file for the lesson and set up your workspace to use an optimal layout for taking lessons.

1. To open your start file, in Flash select File > Open and navigate to the file:
 - On Windows 2000 or XP, browse to *boot drive*\Documents and Settings\All Users\Application Data\Macromedia\Flash MX 2004\language\Configuration\ HelpPanel\HowDoI\BasicFlash\start_files and double-click layers_start fla.

Note: If the Application Data folder is hidden, you'll need to change your Windows Explorer settings to see the folder.

 - On Windows 98, browse to *boot drive*\Windows\Application Data\Macromedia\Flash MX 2004\language\Configuration\HelpPanel\HowDoI\BasicFlash\start_files and double-click layers_start fla.
 - On Macintosh, browse to *Macintosh HD*/Users/Shared/Application Support/Macromedia/Flash MX 2004/language/Configuration/HelpPanel/HowDoI/BasicFlash/start_files and double-click layers_start fla.

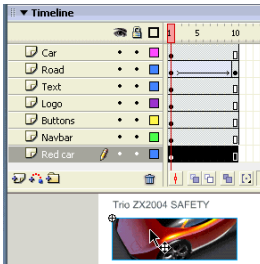
Note: The BasicFlash\finished_files folder contains completed versions of lesson FLA files, for your reference.
2. Select File > Save As and save the document with a new name, in the same folder, to preserve the original start file.
3. Select Window > Panel Sets > Training Layout to modify your work area for taking lessons.
4. In the Stage View pop-up menu, in the upper right side of the Timeline, select Show Frame to view both the Stage and the work area.
5. If necessary, drag the lower edge of the Timeline (Window > Timeline) down to enlarge the Timeline view.

You can also use the scroll bar to scroll through the layers.

Select a layer

You place objects, add text and graphics, and edit on the active layer. To make a layer active, you either select the layer in the Timeline or select a Stage object in the layer. The active layer is highlighted in the Timeline, and the pencil icon indicates it can be edited.

1. In the Tools panel, click the Selection tool.
2. On the Stage, select the red car.



A pencil icon in the Timeline indicates that the Red Car layer is now the active layer.

3. Select the Text layer in the Timeline.

The text blocks above and below the red car are both selected on the Stage, since they're both on the Text layer.

Hide and show layers

You can hide layers to view content on other layers. When hiding layers, you have the option of hiding all layers in your document simultaneously or hiding layers individually.

1. Click the eye icon above the layers, so that a red X appears in the Eye column.

All content disappears from the Stage.

2. One by one, click each red X in the column and watch the content for the layer reappear on the Stage.

Controls to the right of each layer name let you show or hide the contents of a layer.

Note: You may need to use the scroll bar to see all the layers.

Lock a layer

When you've placed content as desired on a layer, you can lock the layer to avoid inadvertent changes to the content by you or by others working on the document.

1. In the Timeline, click the black dot under the Lock column, next to the Logo layer.

A padlock icon appears, indicating the layer is now locked.


2. With the Selection tool, try to drag the logo that appears along the top of the Stage.

You can't drag the logo, because the layer is locked.

Note: If you accidentally drag something from an unlocked layer, press Control+Z (Windows) or Command+Z (Macintosh) to undo your change.

Add and name a layer

You'll now add a layer, name the layer, and then add a graphic symbol to the layer.

1. In the Timeline, click the Car layer.
2. Click the Insert Layer button below the Timeline.
 The new layer appears above the Car layer and becomes the active layer.
3. Double-click the layer name, type **Background** as the new name for the layer, and press Enter (Windows) or Return (Macintosh).
As a best practice, always name each layer, and give the layer a meaningful name that indicates the type of content on the layer.
4. In the Library panel (Window > Library), select the background graphic symbol and drag it to the Stage.
Because the Background layer is above all layers except the Mask layer, objects on that layer appear over objects on lower layers.


Change the order of layers

Obviously, you don't want the background to cover the other objects on the Stage. Normally, the background layer is the bottom layer on the Timeline. You'll move the Background layer you just created.

1. In the Timeline, drag the Background layer from the top position to the bottom position.
All objects on the Stage now appear on top of the background.
2. With the Background layer still selected, in the Property inspector, enter **0** in the X text box and **72** in the Y text box. Press Enter or Return to precisely position the Background layer on the Stage.

Organize layers in a folder

You can create layer folders to organize layers and reduce Timeline clutter. The Timeline contains two layers that contain navigation objects: one for navigational buttons and another for navigational art. You'll create a layer folder, named Navigation, for both layers.

1. In the Timeline, select the Buttons layer.
 2. Click the Insert Layer Folder button, which is below the layer names.
 **Note:** If the Property inspector shows properties for the frame rather than for the movie clip, click the Background movie clip on the Stage.
 3. Double-click the layer folder name and rename the folder **Navigation**.
 4. Drag the Navbar layer and the Buttons layer to the Navigation folder.
The layers appear indented to indicate that they're within the folder.
- You can click the expander arrow to expand and collapse the folder and included layers.

Add a mask layer

Using a mask layer provides a simple way to selectively reveal portions of the layer or layers below it. Masking requires making one layer a mask layer and the layers below it masked layers.

You'll use the rectangular shape on the Stage to mask part of the road graphic and animation so that the animation fits better on the Stage.

1. On the Stage, with the Selection tool selected, click the rectangular shape below the road.
2. Drag the shape straight up and align the left edge of the shape with the left edge of the road.
3. Right-click (Windows) or Control-click (Macintosh) the Mask layer in the Timeline and select Mask from the context menu.

The layer converts to a mask layer, indicated by a blue diamond-shaped icon. The layer immediately below the layer is linked to the mask layer. The masked layer's name is indented, and its icon changes to a blue layer icon.

4. In the Timeline, drag the Road layer to the Mask layer, placing it below the Car layer.

The mask layer and the layers it masks are automatically locked.

5. To view the mask effect, select Control > Test Movie.
6. When you finish viewing the mask effect, close the SWF file window to return to your document.

Add a guide layer

So far you've learned about regular layers and mask layers. The third type of layer is a guide layer. You use guide layers to hold content that you don't want to appear in your published or exported file. For example, you could place instructions to others working on your document on a guide layer. As you take lessons in Flash, you'll notice that many of the lesson FLA files contain placeholders, which indicate where to place a Stage object, on guide layers. You'll create a guide layer now.

1. In the Timeline, select the Background layer, then click the Insert Layer button to create a new layer.
2. Name the new layer **Notes** and press Enter or Return.
3. Right-click (Windows) or Control-click (Macintosh) the Notes layer and select Guide from the context menu.

An icon next to the layer name indicates that the layer is a guide layer.

4. With the Notes layer still selected, click the Text tool in the Tools panel. Then, in the area of the Stage above the car and road, enter **Production note: Animation with no stop (); actions loop by default.**
5. Save your document and select Control > Test Movie.

No content that you added to the guide layer appears in the SWF file window.

6. When you finish viewing the SWF file, close the window to return to the document.

Delete a layer

Since you don't really need the guide layer in your document, you'll delete it.

- In the Timeline, with the Notes layer selected, click the Delete Layer button.

Summary

Congratulations on learning how to work with layers in Flash. In just a few minutes, you learned how to accomplish the following tasks:

- Select a layer
- Hide and show layers
- Lock a layer
- Add and name a layer
- Change the order of layers
- Organize layers in a folder
- Add a mask layer
- Add a guide layer
- Delete a layer

To learn more about Flash, take another lesson.

CHAPTER 6

Create a User Interface with Layout Tools

Macromedia Flash MX 2004 and Macromedia Flash MX Professional 2004 offer a variety of ways to place objects precisely on the Stage, letting you choose your preferred method. In this lesson, you'll use layout tools to create a user interface. Specifically, you'll accomplish the following tasks:

- View the workspace rulers
- Use guides to align objects
- Change the Stage size
- Resize objects to match the Stage size
- Align an object using the alignment guides
- Align objects using the Align panel
- Snap objects to each other
- Align objects using the Property inspector
- Use the grid and arrow keys to align objects

Although this lesson uses specific tools for particular types of objects (alignment guides to align text, for example), no strict rules exist about the best way to align a particular type of object. In your own projects, use whatever tools suit your needs.

Set up your workspace

First, you'll open the start file for the lesson and set up your workspace to use an optimal layout for taking lessons.

1. To open your start file, in Flash select File > Open and navigate to the file:
 - On Windows 2000 or XP, browse to *boot drive*\Documents and Settings\All Users\Application Data\Macromedia\Flash MX 2004\language\Configuration\HelpPanel\HowDoI\BasicFlash\start_files and double-click layout_tools_start fla.

Note: If the Application Data folder is hidden, you'll need to change your Windows Explorer settings to see the folder.

 - On Windows 98, browse to *boot drive*\Windows\Application Data\Macromedia\Flash MX 2004\language\Configuration\HelpPanel\ HowDoI\BasicFlash\start_files and double-click layout_tools_start fla.
 - On Macintosh, browse to *Macintosh HD*/Users/Shared/Application Support/Macromedia/Flash MX 2004/language/Configuration/HelpPanel/HowDoI/BasicFlash/start_files and double-click layout_tools_start fla.

Note: The BasicFlash\finished_files folder contains completed versions of lesson FLA files, for your reference.
2. Select File > Save As and save the document with a new name, in the same folder, to preserve the original start file.
3. Select Window > Panel Sets > Training Layout to modify your workspace for taking lessons.
4. In the Stage View pop-up menu, in the upper right of the Timeline, select Show Frame to view both the Stage and the work area.
5. Click in the work area, away from objects on the Stage, so that no objects are selected.

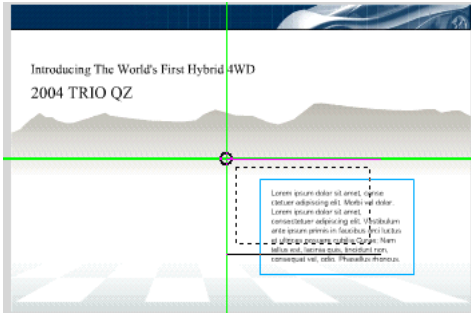
Use guides to align an object

You can use rulers and guides to precisely position or align objects in your document. You'll add guides to help you center the block of text.

1. Select View > Rulers.

A horizontal and vertical ruler appear above and to the left of the Stage.
2. Click anywhere in the horizontal ruler and drag down until you reach the 250-pixel horizontal position.
3. Click anywhere in the vertical ruler and drag left until you reach the 375-pixel vertical position.
4. In the Tools panel, click the Selection tool.

5. On the Stage, click the upper left corner of the text block and drag it to the intersection of the two guides.



A small circle appears when the selection snaps to the guides.

6. If you want to remove the guides, select View > Guides > Clear Guides.

Change the Stage size

The Stage size of your document is 750 pixels x 500 pixels. You'll change the Stage size to 640 x 480, a common size that supports a wide variety of screen sizes and resolutions.

1. Click in a blank area of the work area to deselect the text block.
2. In the Property inspector, you see properties for the entire document. Click the Size button.
3. In the Document Properties dialog box, enter **640** for the width and **480** for the height, and click OK.

The document size changes, but the objects on the Stage remain the same size.

Resize objects to match the Stage size

When you changed the Stage size, art on the Stage extended off the Stage and into the work area. You can easily resize the art to match the Stage size again.

1. On the Stage, click the gray mountains to select them. Shift-click the blue banner at the top of the Stage to add it to the selection.

2. Open the Align panel (Window > Design Panels > Align).

Tooltips appear in the Align panel showing the names of alignment options.

3. In the Align panel, select To Stage; then, under Match Size, select Match Width. The size of the selected art changes to match the width of the Stage.

4. Still in the Align panel, click Align Left Edge.

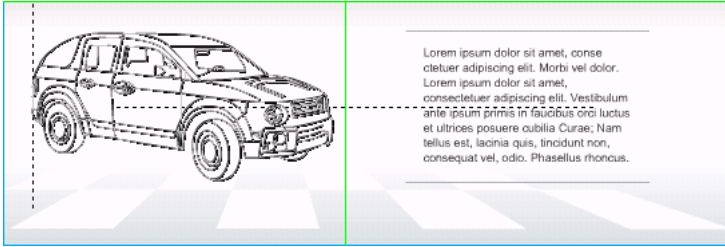
The art aligns to the left edge of the Stage.

5. In the Timeline, click the Bevel layer to select it.

6. From the Library panel (Window > Library), drag the bevel graphic to any area toward the bottom of the Stage.

7. In the Align panel, verify that To Stage is still selected, and click Match Width.

5. While keeping the car centered with the body text (you should continue to see the center alignment guide), drag the auto in a straight line, toward the left of the Stage, until the guide snaps at the 30-pixel border that you created previously.



You've aligned the car with the text, as well as to the snapping border.

Align objects using the Align panel

Previously, you used the Align panel to match objects to the size of the Stage. Now you'll use the Align panel to center objects in relation to each other, and then in relation to the Stage.

1. With the Selection tool, click the text on the Stage that reads "Introducing the World's First Hybrid 4WD."
2. Press the Shift key and click the second line of title text, "2004 Trio QZ," to add it to the selection.
3. In the Align panel (Window > Design Panels > Align), deselect To Stage and select Align Horizontal Center.

You've centered the two rows in relation to their horizontal axes. Next, you'll group the title text to center both lines in relation to the Stage.

4. With both lines of text still selected, select Modify > Group.
5. In the Align panel, select To Stage, then select Align Horizontal Center again.
With To Stage selected, the objects align in relation to the horizontal center of the Stage.
6. Close the Align panel.

Snap objects to each other

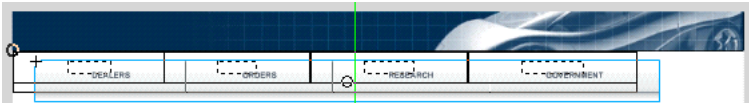
You can snap objects on the Stage to other objects on the Stage, thereby setting the alignment of objects to each other. Using the Snap to Objects feature, you'll align a navigation bar with the top Stage banner.

1. Select View > Snapping. In the submenu, select Snap to Objects if it's not already selected.
2. In the Timeline, select the Nav layer.

3. From the Library panel, drag the nav graphic to any area of the Stage below the blue banner, and then release the nav graphic.



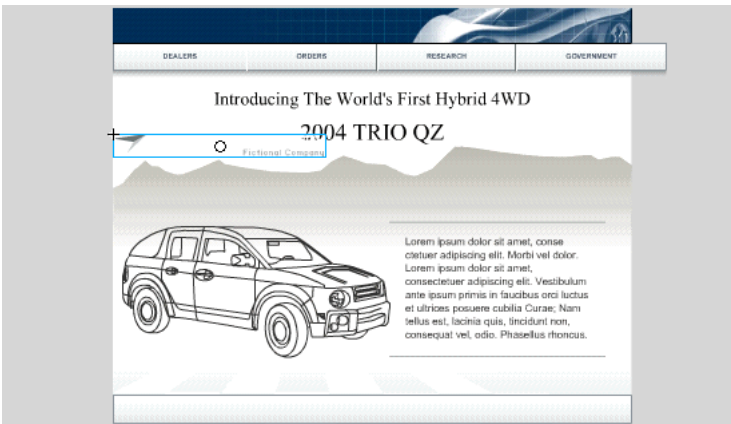
4. Click the upper left corner of the nav graphic, then drag it so that a circular snap indicator appears.
5. With the snap indicator visible, drag the upper left corner of the nav bar and snap it into place against the lower left corner of the blue banner.



Align objects using the Property inspector

The Property inspector lets you precisely align objects on the x and y Stage axes, from the registration point of the Stage object. The registration point is the point from which a symbol aligns or rotates. You'll use the Property inspector to align the logo.

1. In the Timeline, select the Top layer.
2. From the Library panel (Window > Library), drag the logo to an empty area of the Stage.



3. In the Property inspector, with the logo still selected, enter **20** in the X text box and **8** in the Y text box. Press Enter or Return.

The logo moves to the new x and y Stage values.

Note: You can view and change the registration point of an object in the Info panel (Window > Design Panels > Info). The black square in the grid represents the registration point. To change it, you click another square in the grid.

Align objects using the grid and arrow keys

You can use the grid to assist you in placing objects on the Stage.

1. Select View > Grid > Show Grid.

The grid does not appear when you test or publish your document.

Note: If you wanted to snap objects to the horizontal and vertical grid lines, you would also select Snap to Grid (View > Snapping > Snap to Grid). For this lesson, you won't snap objects to the grid.

2. On the Stage, select the title text you previously grouped together.
3. Use the Up Arrow key on your keyboard to nudge the text until the first line in the title text is on a horizontal grid line. Be sure to leave space between the title text and the navigation bar.

Note: You can also use the Left Arrow, Down Arrow, and Right Arrow keys to nudge objects on the Stage in the direction of the arrow.

Summary

Congratulations on using layout tools to create a user interface. In a few minutes, you learned how to accomplish the following tasks:

- View the workspace rulers
- Use guides to align objects
- Change the Stage size
- Resize objects to match the Stage size
- Align an object using the alignment guides
- Snap objects to each other
- Align objects using the Property inspector
- Use the grid and arrow keys to align objects

For more information about design topics in Flash, take another lesson from the Basic Flash series.

CHAPTER 7

Draw in Flash

When you draw in Flash, you create vector art, which is a mathematical representation of lines, curves, color, and position. Vector art is resolution-independent; you can rescale the art to any size or display it at any resolution without losing clarity. Additionally, vector art downloads faster than comparable bitmap images. In this lesson, you'll create vector art of a bolt and logo as you accomplish the following tasks:

- Create a polygon
- Rotate a shape
- Cut out a shape within a shape
- Transform artwork
- Copy strokes
- Draw lines with the Line tool
- Select and add a fill color
- Group a shape
- Create a logo with the Pen tool

Set up your workspace

First, you'll open the start file for the lesson and set up your workspace to use an optimal layout for taking lessons.

1. To open your start file, in Flash select File > Open and navigate to the file:
 - On Windows 2000 or XP, browse to *boot drive*\Documents and Settings\All Users\Application Data\Macromedia\Flash MX 2004\language\Configuration\HelpPanel\HowDoI\BasicFlash\start_files and double-click drawing_start fla.
 - On Windows 98, browse to *boot drive*\Windows\Application Data\Macromedia\Flash MX 2004\language\Configuration\HelpPanel\HowDoI\BasicFlash\start_files and double-click drawing_start fla.
 - On Macintosh, browse to *Macintosh HD*/Users/Shared/Application Support/Macromedia/Flash MX 2004/language/Configuration/HelpPanel/HowDoI/BasicFlash/start_files and double-click drawing_start fla.

Note: The BasicFlash\finished_files folder contains completed versions of lesson FLA files, for your reference.

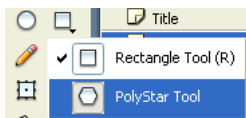
Flash opens in the authoring environment.

2. Select File > Save As and save the document with a new name, in the same folder, to preserve the original start file.
As you complete this lesson, remember to save your work frequently.
3. Select Window > Panel Sets > Training Layout to configure your workspace.

Select a shape tool

Shape tools offer an easy way to create figures such as ovals, rectangles, polygons, and stars. You'll use the PolyStar tool to create a polygon.

1. In the Timeline, select the Content layer.
2. In the Tools panel, select the PolyStar tool. You may need to click the lower right control on the Rectangle tool to see a menu that displays the PolyStar tool.



3. Click anywhere in the gray work area beside the Stage to display properties for the shape you'll create. In the Property inspector (Window > Properties), verify that black is selected as the stroke color, 1 pixel is selected as the stroke height, and Solid is selected as the stroke style.

The stroke is the line that outlines your shape.

4. Click the Fill Color control and select blue with a hexadecimal value of #0000FF.

The fill color appears within the stroke in a shape.

Select options to create a polygon

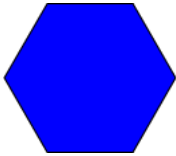
The PolyStar tool offers options that let you specify the number of sides in a polygon; you can also use options for the tool to create a star. You'll specify that your polygon have six sides.

1. In the Property inspector, with the PolyStar tool still selected, click Options.
2. In the Tools Setting dialog box, verify that Polygon appears in the Style pop-up menu, and enter 6 in the Number of Sides text box. Click OK.

Draw a polygon

The Shift key assists you in constraining your shape along a vertical or horizontal line.

- Press Shift and drag the mouse over the left side of the Stage (away from the numbers on the Guides layer) to draw a hexagon, as shown in the following illustration:



Rotate the shape

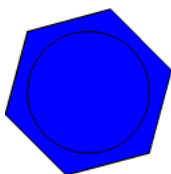
Once you've created a shape, you can use the Transform panel to specify a precise number of degrees in which to rotate the shape.

1. In the Tools panel, click the Selection tool. On the Stage, double-click within the hexagon to select both the stroke and the fill.
Clicking just once within a shape selects only the fill.
2. Select Window > Design Panels > Transform. In the Transform dialog box, verify that Rotate is selected, and enter -15 in the Rotate text box to rotate the shape 15° clockwise. Press Enter or Return.

Use the cut-out feature

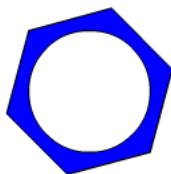
When you create one shape on top of another on the same layer, and the two shapes are ungrouped, the shape on top “cuts out” the area of the shape underneath. You’ll create a circle within the hexagon, then cut out the circle.

1. Select View > Snapping and select Snap to Objects if it’s not already selected.
2. In the Tools panel, click the Oval tool. While pressing Shift to constrain the shape, draw a circle within the hexagon (imagine the hexagon as the face of a clock and begin your circle at the 10:00 corner point, dragging to the 4:00 corner point), as in the following illustration.



Note: If you make an error drawing the circle, press Control+Z (Windows) or Command+Z (Macintosh) to undo your circle.

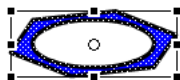
3. In the Tools panel, click the Selection tool. On the Stage, click within the circle and press Backspace or Delete.



Transform the shape of your drawing

Using the Free Transform tool you can scale, rotate, compress, stretch, or skew lines and shapes. You’ll use the Free Transform tool to compress your drawing.

1. In the Tools panel, select the Free Transform tool. Double-click the hexagon on the Stage to select both the stroke and the fill.
2. Drag the top middle handle of the Free Transform tool down to transform the hexagon into the following shape:



Copy strokes

You can select and copy strokes, which you'll do now to create the lower edge of the bolt.

1. With the Selection tool, click anywhere on the Stage or work area, away from an object, to deselect the shape.
2. Hold Shift and click the three lines of the shape on the Stage that comprise the bottom of the hexagon to select them, as shown in the following illustration:



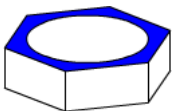
3. Press Shift + Alt and drag down slightly to drag a copy of the three lines, as in the following illustration:



Draw with the Line tool

The Line tool allows you to draw straight lines in any direction.

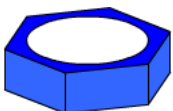
- In the Tools panel, select the Line tool. On the Stage, draw four vertical lines from the hexagon to the stroke copies that you dragged down, as shown in the following illustration:



Select and add a different fill color

You can use the Paint Bucket tool to change an existing color and to fill empty areas surrounded by lines. You'll use the Paint Bucket tool to add a fill color to the empty areas of your drawing.

1. In the Tools panel, select the Paint Bucket tool. In the Colors area, click the Fill Color control and select blue with the hexadecimal value of #3366FF.
2. On the Stage, click within the lines to add the selected color to the empty areas:



Group the shape

You can manipulate the stroke and fill of a shape as separate entities, as you did earlier, or you can group the stroke with the fill to manipulate the shape as a single graphic, which you'll do now.

1. With the Selection tool, drag around the shape to select both the stroke and the fill. Select **Modify > Group**.
2. Click the fill area and move the shape around the left side of the Stage, as desired, to place it.

Create a logo with the Pen tool

The Pen tool offers a way to draw precise straight or curved line segments. You click to create points on straight line segments, and click and drag to create points on curved line segments. You can adjust the segments by adjusting points on the line. You'll use the Pen tool to create the logo.



1. In the Tools panel, select the Pen tool.
2. Click the dot next to the number 1, and then click the dot next to the number 2 to create a straight line segment.



3. Click the dot next to the number 3 and next to the number 4 in sequence (as though you were completing a dot-to-dot drawing) to create the straight line segments that comprise the logo.
4. To close the path, position the Pen tool over the first anchor point (the dot by number 1). A small circle appears next to the pen tip when it is positioned correctly. Click to close the path. Once the path is closed, it fills with the selected fill color.

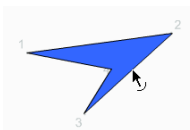
By default, selected curve points appear as hollow circles, and selected corner points appear as hollow squares.

5. With the Selection tool, move the pointer around the logo that you created.

When you move the pointer over a corner point, which you can click and drag to create various angles, the pointer appears as follows:



When you move the pointer over a curved line segment (you don't have curve points in your logo), which you can click and drag to create various curved lines, the pointer appears as follows:



Summary

Congratulations on learning how to use some of the various drawing tools in Flash. In a few minutes, you learned how to accomplish the following tasks:

- Create a polygon
- Rotate a shape
- Cut out a shape within a shape
- Transform artwork
- Copy strokes
- Draw lines with the Line tool
- Select and add a fill color
- Group a shape
- Create a logo with the Pen tool

To learn more about creating art in Flash, see “Drawing” in *Using Flash*.

CHAPTER 8

Create Symbols and Instances

A symbol is a reusable object, and an instance is an occurrence of a symbol on the Stage. Repeated use of instances does not increase the file size and is a good part of a strategy for keeping a document file size small. Symbols also simplify editing a document; when you edit a symbol, all instances of the symbol update to reflect the edits. Another benefit of symbols is that they allow you to create sophisticated interactivity.

In this lesson you'll work with symbols and instances to accomplish the following tasks:

- Create a graphic symbol
- Duplicate and modify an instance
- Create a movie clip symbol
- Edit a symbol by adding an effect

Set up your workspace

First, you'll open the start file for the lesson and set up your workspace to use an optimal layout for taking lessons.

1. To open your start file, in Flash select File > Open and navigate to the file:
 - On Windows 2000 or XP, browse to *boot drive*\Documents and Settings\All Users\Application Data\Macromedia\Flex MX 2004\language\Configuration\HelpPanel\HowDoI\BasicFlash\start_files and double-click symbols_start fla.
 - On Windows 98, browse to *boot drive*\Windows\Application Data\Macromedia\Flex MX 2004\language\Configuration\HelpPanel\HowDoI\BasicFlash\start_files and double-click symbols_start fla.
 - On Macintosh, browse to *Macintosh HD*/Users/Shared/Application Support/Macromedia/Flex MX 2004/language/Configuration/HelpPanel/HowDoI/BasicFlash/start_files and double-click symbols_start fla.

Note: The BasicFlash\finished_files folder contains completed versions of lesson FLA files, for your reference.

The document opens in the Flash authoring environment.

2. Select File > Save As and save the document with a new name, in the same folder, to preserve the original start file.
As you complete this lesson, remember to save your work frequently.
3. Select Window > Panel Sets > Training Layout to configure your workspace.

About creating symbols

When you create a symbol, you specify one of the following symbol behaviors:

- Graphic
- Movie clip
- Button

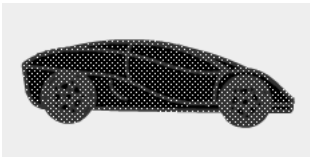
In this lesson, you'll work with graphic and movie clip symbols. For a lesson about button symbols, select Help > How Do I > Basic Flash > Add Animation and Navigation to Buttons.

Create a graphic symbol

A graphic symbol is well suited for repeated use of static images, or for creating animations associated with the main Timeline. Unlike with movie clip and button symbols, you cannot give instance names to graphic symbols, nor can you refer to them in ActionScript.

You'll take vector art on the Stage and turn it into a graphic symbol.

1. In the Tools panel, click the Selection tool.
2. On the Stage, drag around the car to select it.



3. Select Modify > Convert to Symbol.
4. In the Convert to Symbol dialog box, enter **CarGraphic** as the name and select Graphic as the behavior.
5. The Registration grid uses a small black square to indicate where, within the symbol bounding box, the registration point is located. A registration point is the axis around which the symbol rotates, and the point along which the symbol aligns. Click the upper left square on the grid to select the registration point location, and click OK.
6. The car on the Stage is now an instance of the CarGraphic symbol. The Property inspector shows properties for the graphic symbol instance.



7. Open the Library panel (Window > Library) to view the symbol.
You'll find the CarGraphic symbol in the Library panel. Flash stores symbols in the library. Each document has its own library, and you can share libraries between different FLA files.

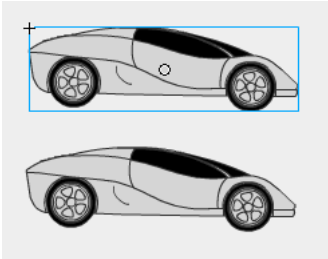
Duplicate and modify an instance of a symbol

Once you've created a symbol, you can use instances of it repeatedly in your document. You can modify the following instance properties for an individual instance without affecting other instances or the original symbol: color, scale, rotation, alpha transparency, brightness, tint, height, width, and location.

If you edit the symbol later, the instance retains its modified properties in addition to acquiring the symbol edits.

You'll now duplicate the instance of the car, then change the tint for the duplicate.

1. On the Stage, select the car. Press Alt and drag the car up to create another instance.



2. With the duplicate selected, select Tint from the Color pop-up menu in the Property inspector.
3. In the RGB area, enter 0 in the Red Color pop-up menu, 0 in the Green Color pop-up menu, and 255 in the Blue Color pop-up menu. Then press Enter or Return.

The duplicate instance turns blue, but the original instance remains unchanged.

Modify a symbol

You can enter symbol-editing mode by double-clicking any instance of a symbol. Changes that you make in symbol-editing mode affect all instances of the symbol.

1. Do one of the following to enter symbol-editing mode:
 - On the Stage, double-click one of the car instances.
 - In the Library panel, double-click the CarGraphic symbol.

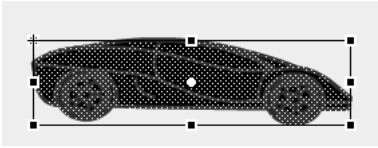
Next to Scene 1 toward the top of the workspace, the name of the symbol appears, which indicates that you're in symbol-editing mode for the named symbol.



2. In the Tools panel, select the Free Transform tool and drag around the bottom car to select the entire car.

In symbol-editing mode, the car is a graphic, within a symbol, that you can manipulate as you would any other vector graphic.

3. Drag the middle-right sizing handle of the Free Transform tool slightly to the right, to stretch the symbol.



4. Click Scene 1, above the Timeline, to exit symbol-editing mode.
Both instances of the symbol reflect the transformation.

Create a movie clip symbol

A movie clip symbol is analogous in many ways to a document within a document. This symbol type has its own Timeline independent of the main Timeline. You can add movie clips within other movie clips and buttons to create nested movie clips. You can also use the Property inspector to assign an instance name to an instance of a movie clip, then reference the instance name in ActionScript.

You'll convert the tire on the Stage into a movie clip.

1. With the Selection tool, click the tire on the Stage to select it and select **Modify > Convert to Symbol**.
2. In the **Convert to Symbol** dialog box, enter **MCWheel** as the name, and select **MovieClip** as the behavior.
3. In the Registration grid, this time select the center square as the registration point, so the center of the movie clip becomes the axis around which the symbol rotates. Then click **OK**.

The image on the Stage is now an instance of the **MCWheel** symbol in the library.

Assign an instance name to the movie clip

To refer to an instance in ActionScript, and as a general best practice, always assign instance names to buttons and movie clip symbols. (You cannot assign an instance name to a graphic symbol.)

- In the Property inspector, with the instance of **MCWheel** selected on the Stage, enter **wheel_mc** in the Instance Name text box.

Add an effect to the movie clip

You can create an animation within a movie clip Timeline, in symbol-editing mode, that plays independent of the main Timeline. You'll add an effect to the MCWheel symbol that will cause all instances of the symbol to spin.

1. With the Selection tool, double-click the wheel_mc instance to enter symbol-editing mode.
2. Right-click (Windows) or Control-click (Macintosh) the symbol and select Timeline Effects > Transform/Transition > Transform.
3. In the Transform dialog box, enter **60** in the Effect Duration text box to specify that the effect span 60 frames in the Timeline.
4. In the Spin text box, enter **1** and verify that 360 populates the Rotate text box.
5. Click Update Preview to view a preview of the effect, then click OK.

The Effect spans 60 frames in the movie clip Timeline.

6. Click Scene 1 above the Timeline to exit symbol-editing mode.
7. Select Control > Test Movie to view the animation.

Summary

Congratulations on learning about symbols and instances. In a few minutes, you accomplished the following tasks:

- Create a graphic symbol
- Duplicate and modify an instance
- Create a movie clip symbol
- Edit a symbol by adding an effect

To learn more about Flash, take another lesson in the Basic Flash series.

CHAPTER 9

Add Animation and Navigation to Buttons

A *button* is a symbol that contains special frames for different button states, such as when the user's mouse pointer is over the button or when the user clicks the button. When you select the Button behavior for a new symbol, Macromedia Flash MX 2004 and Macromedia Flash MX Professional 2004 create the Timeline for the button states. You can add navigation to buttons by using behaviors or by writing ActionScript.

In this lesson, you'll learn about creating and modifying buttons, including adding animation to a button. Specifically, you'll learn how to complete the following tasks:

- Create a button from grouped objects
- Name a button instance
- View the hit area of a button
- Change the hit area of a button
- Align buttons
- Create animation for a button state
- Add an action to a button
- Add navigation to a button

If you are not familiar with symbols and instances, before taking this lesson select Help > How Do I > Basic Flash > Create Symbols and Instances to take that lesson.

Set up your workspace

First, you'll open the start file for the lesson and set up your workspace to use an optimal layout for taking lessons.

1. To open your start file, in Flash select File > Open and navigate to the file:
 - On Windows 2000 or XP, browse to *boot drive*\Documents and Settings\All Users\Application Data\Macromedia\Flash MX 2004\language\Configuration\HelpPanel\HowDoI\BasicFlash\start_files and double-click *buttons_start fla*.

Note: If the Application Data folder is hidden, you'll need to change your Windows Explorer settings to see the folder.

 - On Windows 98, browse to *boot drive*\Windows\Application Data\Macromedia\Flash MX 2004\language\Configuration\HelpPanel\HowDoI\BasicFlash\start_files and double-click *buttons_start fla*.
 - On Macintosh, browse to *Macintosh HD*/Users/Shared/Application Support/Macromedia/Flash MX 2004/language/Configuration/HelpPanel/HowDoI/BasicFlash/start_files and double-click *buttons_start fla*.

Note: The BasicFlash\finished_files folder contains completed versions of lesson FLA files, for your reference.
2. Select File > Save As and save the document with a new name, in the same folder, to preserve the original start file.
3. Select Window > Panel Sets > Training Layout to modify your work area for taking lessons.
4. In the Stage View pop-up menu, in the upper right side of the Timeline, select Show Frame to view both the Stage and the work area.
5. Click in the work area, away from objects on the Stage, so that no objects are selected.

Create a button from grouped objects

You can create buttons from text and graphics, including bitmap images and grouped objects. In this lesson, you'll turn a logo and some text into one large button.

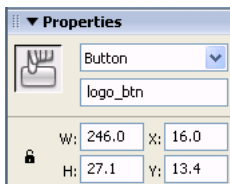
1. In the Tools panel, click the Selection tool. On the Stage, select the grouped text and logo, and then select Modify > Convert to Symbol.
2. In the Convert to Symbol dialog box, name the symbol **BTNLogo**, and select Button as the behavior.
3. In the Registration grid, verify that the square in the upper left corner is selected as the registration point, and click OK.

The registration point is the point from which the symbol aligns and rotates.

Name a button instance

As a best practice, name instances of symbols on the Stage. ActionScript relies on the instance name to identify the object.

- With the button that you created still selected, open the Property inspector (Window > Properties). In the Instance Name text box, type **logo_btn**.



View the hit area by enabling buttons

When the Enable Simple Buttons feature is active, you can view the less complex aspects of your buttons, such as the hit area (the clickable area of a button) and the colors used for the button states. More complex button design, such as animation, does not play.

1. Click in an empty space in the work area, to make sure you don't have any objects selected.
2. Select Control > Enable Simple Buttons, and then move the mouse pointer over different areas of the button that you created.

The only areas of the button that are clickable (as indicated when the pointer changes to a hand) are the text area and the logo. The white space around the text and logo are not clickable.



3. Select Control > Enable Simple Buttons again to deselect the feature so that you can edit the button.

Next, you'll redefine the hit area so that the button area covers the entire grouped area.

Change the hit area of a button

You can specify a hit area that's a different size from the default hit area by adding a keyframe to the Hit frame of the button symbol, and then drawing a shape that defines the hit area.

1. On the Stage, double-click the logo button that you created to view the button Timeline. The button Timeline contains the following states:
 - Up
 - Over
 - Down
 - Hit
2. Double-click the Layer 1 name in the Timeline and rename the layer **Hit Area**.

3. Select the Hit frame (Frame 4) of the Hit Area layer in the BTNLogo Timeline, and press F6 to add a keyframe.
4. In the Tools panel, select the Rectangle tool. The stroke and fill color are unimportant. On the Stage, draw a rectangle that covers, as closely as possible, the logo and text.



The rectangle now defines the clickable area of the button.

5. Click Scene 1, above the upper left side of the Stage, to exit symbol-editing mode for the button.
6. Select Control > Enable Simple Buttons.
7. On the Stage, again move the pointer over the text.
The hit area changes to the shape of the rectangle that you drew.
8. Select Control > Enable Simple Buttons to deselect that feature.

Align buttons

You can align buttons along horizontal and vertical axes using the Align panel.

1. In the Timeline, click the Content layer.
2. Click the Insert Layer button below the Timeline.
3. Double-click the layer name, type **Animated Buttons** as the new name for the layer, and press Enter or Return.
4. In the Library panel (Window > Library) select Button 1 and drag it to the lower right edge of the Stage. Precise placement isn't necessary.



5. In the Instance Name text box in the Property inspector, give the button the instance name **links_btn**.
6. Drag Button 2 and Button 3 from the Library panel, placing them to the left of Button 1. Use the approximate spacing shown in the following illustration:



7. Using the Selection tool, drag to select all three buttons.
8. Open the Align panel by selecting Window > Design Panels > Align. Verify that To Stage is not selected, because you do not want to align the buttons relative to the Stage.
9. In the Align panel, click Align Vertical Center, and then click Distribute Horizontal Center.
The buttons align on the Stage.
10. Close the Align panel.
11. On the Stage, click in an empty space in the work area to make sure you don't have any objects selected, then select Button 2. In the Instance Name text box in the Property inspector, enter **contact_btn**. Select Button 3 and name it **sweepstakes_btn**.

Create animation for a button state

You'll create a movie clip within the Over state of Button 1, and then create a shape tween in the movie clip. The shape tween creates an effect that changes the color from gray to red.

1. On the Stage, double-click Button 1 to open it in symbol-editing mode.
2. In the Button 1 Timeline, hide all layers except the Color layer. In the Color layer, select the Over keyframe.

Note: You hide layers by clicking the dot under the Eye column next to the layer name so that a red X appears.

3. On the Stage, select the black oval shape for Button 1. Press F8 to make the oval a symbol.
4. In the Convert to Symbol dialog box, name the symbol **Button Animation**. Select Movie Clip (not Button) as the behavior, and click OK.
5. On the Stage, double-click the Button Animation symbol to switch to symbol-editing mode.
6. Rename Layer 1 **Color Change**. Select Frame 15, and then press F6 to add a keyframe.
7. With the playhead still on Frame 15, select the button shape on the Stage, and in the Property inspector select a bright shade of red from the Fill Color pop-up menu.
8. In the Timeline, click any frame between Frames 1 and 15. In the Property inspector, select Shape from the Tween pop-up menu.

Drag the playhead from Frames 1 to 15 to see the color change.

Add an action to a button

When the user clicks the button and the tweened animation plays, you want the playhead to move to the end of the Button Animation Timeline and then stop. You'll use ActionScript to control playhead movement in a Timeline.

1. Add a new layer to the Button Animation Timeline and name it **Actions**.
2. On the Actions layer, add a keyframe to Frame 15 by pressing F6.
3. Open the Actions panel (Window > Development Panels > Actions) and if necessary, enlarge it to view both the Actions toolbox and the Script pane.
4. With Frame 15 of the Actions layer selected, go to the Global Functions > Timeline Control category of the Actions toolbox and double-click `stop`.

The `stop` action lets you specify that the playhead stop when it reaches Frame 15.

In the Button Animation Timeline, Frame 15 of the Actions layer now displays a small *a*, which indicates that an action is attached to that frame.

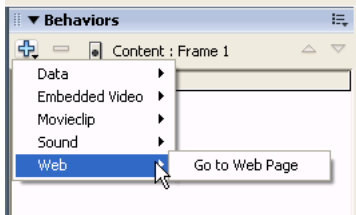
5. Click Scene 1, above the Stage, to exit symbol-editing mode and return to the main document.
6. Click the options menu control in the upper right of the Actions panel, and select Close Panel to close the panel.
7. Select Control > Enable Simple Buttons so you can test the animated button.

8. On the Stage, move the pointer over the button and click the button.
9. Select Control > Enable Simple Buttons to deselect that feature.

Add navigation to a button

With behaviors, you can quickly add navigation to a button without having to know ActionScript. You'll add navigation to open a web page when the user clicks a button.

1. On the Stage, select the instance of Button 1.
2. In the Behaviors panel (Window > Development Panels > Behaviors), click the Add Behavior button and select Web > Go to Web Page.



3. In the Go to URL dialog box, select `_blank` in the Open In pop-up menu to open the URL in a new browser window. In the URL text box, either accept the default setting of `http://www.macromedia.com` or enter a different URL. Then click OK.
4. If desired, repeat the steps above, selecting Button 2, then Button 3, to add navigation to those buttons as well.
5. Click the options menu control in the upper right of the Behavior panel, and select Close Panel to close the panel.

Test the SWF file

You'll test your document to view the button animation and see if the navigation works as expected.

1. Save your document and select Control > Test Movie.
2. Roll the mouse pointer over the instance of Button 1 to view the color animation that you created.
3. Click the button to see if your web browser opens to the URL that you specified.
4. If you added navigation to the other two buttons, test those buttons as well.
5. When you finish viewing the SWF file, close the SWF file and web browser windows.

Summary

Congratulations on learning about buttons. In a few minutes, you were able to accomplish the following tasks:

- Create a button from grouped objects
- Name a button instance
- View the hit area of a button
- Change the hit area of a button
- Align buttons
- Create animation for a button state
- Add an action to a button
- Add navigation to a button

To learn more about Flash, take another lesson.

CHAPTER 10

Add Static, Input, and Dynamic Text

Macromedia Flash MX 2004 and Macromedia Flash MX Professional 2004 provide a variety of text features and options. This lesson introduces you to the three primary types of text you can add to a document. You can add *static text* for titles, labels, or other text content you want to appear in a document. You can also use *input text* options to allow viewers to interact with your Flash application—for example, to enter their name or other information in a form. The third type of text is *dynamic text*. You use dynamic text fields to display text that changes based on criteria you specify. For example, you might use a dynamic text field to add values stored in other text fields, such as the sum of two numbers.

After taking this lesson, be sure to explore additional text options described in “Working with Text” in *Using Flash*.

In this lesson, you’ll add text and text fields to a document as you accomplish the following tasks:

- Create an expanding-width text block
- Create a fixed-width text block
- Edit text and change font attributes
- Select device fonts
- Add an input text field
- Copy a text field
- Assign instance names to text fields
- Create a dynamic text field
- View ActionScript that links the text field to an external text file
- Set up and check spelling

Set up your workspace

First, you'll open the start file for the lesson and set up your workspace to use an optimal layout for taking lessons.

1. To open your start file, in Flash select File > Open and navigate to the file:

- On Windows 2000 or XP, browse to *boot drive*\Documents and Settings\All Users\Application Data\Macromedia\Flash MX 2004\language\Configuration\HelpPanel\HowDoI\BasicFlash\start_files and double-click text_start fla.

Note: If the Application Data folder is hidden, you'll need to change your Windows Explorer settings to see the folder.

- On Windows 98, browse to *boot drive*\Windows\Application Data\Macromedia\Flash MX 2004\language\Configuration\HelpPanel\HowDoI\BasicFlash\start_files and double-click text_start fla.
- On Macintosh, browse to *Macintosh HD*/Users/Shared/Application Support/Macromedia/Flash MX 2004/language/Configuration/HelpPanel/HowDoI/BasicFlash/start_files and double-click text_start fla.

Note: The BasicFlash\finished_files folder contains completed versions of lesson FLA files, for your reference.

2. To open the start file, in Flash select File > Open. Then do one of the following:

- If you're using a Windows 2000 or XP operating system, browse to *boot drive*\Documents and Settings\username\Local Settings\Application Data\Macromedia\Flash MX 2004\language\Configuration\HelpPanel\HowDoI\BasicFlash\start_files and double-click text_start fla.

Note: If the Application Data folder is hidden, you'll need to change your Windows Explorer settings to see the folder.

- If you're using a Windows 98 operating system, browse to *boot drive*\Windows\Application Data\Macromedia\Flash MX 2004\language\Configuration\HelpPanel\HowDoI\BasicFlash\start_files and double-click text_start fla.
- If you're using a Macintosh operating system, browse to *Macintosh HD*/Users/username/Library/Application Support/Macromedia/Flash MX 2004/language/Configuration/HelpPanel/HowDoI/BasicFlash/start_files and double-click text_start fla.

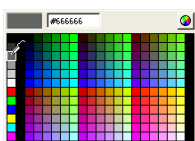
Note: The BasicFlash\finished_files folder contains completed versions of lesson FLA files, for your reference.

3. Select File > Save As and save the document with a new name, in the same folder, to preserve the original start file and to preserve the file's link to dependent files.
4. Select Window > Panel Sets > Training Layout to modify your workspace for taking lessons.
5. In the Stage View pop-up menu, in the upper right side of the Timeline, select Show Frame to view both the Stage and the work area.
6. In the Tools panel, click the Selection tool. Then in the Timeline, click the Static Text layer to select the layer you'll work in.

Create an expanding-width text block

You can define the size of a text block, or you can use a text block that expands to fit the text you write. You'll begin this lesson by simply adding text to a document.

1. Click in a blank area in the workspace to make sure no Timeline frames or objects on the Stage are selected.
2. In the Tools panel, select the Text tool.
3. In the text Property inspector, set the following options:
 - In the Text Type pop-up menu, select Static text, if it isn't already selected.
 - For Font, select Arial.
 - For Font Size, enter 13.
 - Click the text color box and select the gray with a hexadecimal value of #666666.



- Select Align Left, if it isn't already selected.
4. In the Timeline, select the Static Text layer.
 5. With the Text tool still selected, click the Stage along the left edge of Text 1 guide, and type **Trio ZX2004 Safety Features**.

By typing with the Text tool selected, you create a one-line text block that expands as you type. An expanding-width one-line static text block has a round handle in the upper right corner.

Trio ZX2004 Safety Features

6. If necessary, align the text above the Text 1 guide by clicking the Selection tool in the Tools panel and dragging the text that you typed to the guide.

Note: Text 1 guide is on a guide layer, which doesn't appear in your SWF file.

Create a fixed-width text block

In addition to creating a line of text that expands as you type, you can create a text block that has a fixed width. Text that you enter into a fixed-width text block wraps to the next line at the edge of the block.

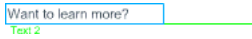
You'll now create a static text block with fixed dimensions.

1. Verify that the Static Text layer is still selected in the Timeline, and use the Selection tool to deselect any objects on the Stage or Timeline frames (click in the workspace, away from objects)
2. In the Tools panel, select the Text tool.
3. In the Property inspector, change the text size to 10 points.

4. On the Stage, drag your pointer over the area of the Text 2 guide.

An extendable one-line static text block has a round handle in the upper right corner, and a fixed-width static text block has a square handle.

5. Inside the text block that you created, type **Want to learn more?**



Want to learn more?
Text 2

Note: You can drag the square handle for a text block to change its width. Additionally, you can double-click a square handle to convert it into a round expanding handle.

6. If necessary, align the text by clicking the Selection tool in the Tools panel and dragging the text that you typed above the Text 2 guide.

Edit text and change font attributes

When you select text, the Property inspector displays standard type formatting features. You can use the Property inspector to change font and size properties for a selected text object.

1. In the Tools panel, click the Selection tool. Double-click the text on the Stage that reads “Submit your contact information.”
2. Select the letter S then type **Please s**, so that the text now reads “Please submit your contact information.”

B

3. In the Tools panel, click the Selection tool. Use the Property inspector to select the bold style.
4. Click the text Fill Color control and choose a different text color, such as another shade of gray.

Select device fonts

When you use a font installed on your system in a Flash document, Flash embeds the font information in the Flash SWF file to ensure that the font is displayed properly. In addition to embedding fonts, you can use the device fonts option. This option finds the fonts on a viewer’s computer that most closely resemble the specified device font. You can use device fonts only with static text.

You’ll specify that the selected text use device fonts.

1. With the “Please submit your contact information” text still selected on the Stage, in the Font pop-up menu in the Property inspector, select `_sans`.
2. In the Property inspector, select Use Device Fonts.

You will not see a difference in the text if you have the Arial font installed on your computer.

Add an input text field

You can use an input text field to allow viewers an opportunity to interact with your Flash application. For example, input text fields offer an easy way to create forms.

In a later lesson (select Help > How Do I > Basic ActionScript > Create a Form with Conditional Logic and Send Data), you'll learn how to use an input text field to send data from Flash to a server. Now, you'll add a text field where viewers will enter their first name.

1. In the Timeline, select the Input Text layer.
2. Use the Selection tool to click in the workspace, away from the objects on the Stage.
3. In the Tools panel, select the Text tool.
4. In the text Property inspector, set the following options:
 - Select Input Text in the Text Type pop-up menu
 - Select Arial in the Font pop-up menu.
 - For the font size, enter **8**.
 - Click the text color box and select a shade of dark blue.



The Alias Text button renders text so that it appears more readable at small sizes. For input text, this option is supported if the end user has Macromedia Flash Player 7 or later.



- Select Single Line in the Line Type pop-up menu, and verify that Show Border Around Text is selected.
- Single Line displays the text as one line. Show Border Around Text indicates the boundaries of the text field with a visible border.
5. On the Stage, drag the pointer in the area to the right of the First Name text to create an input text field.

Please submit your contact information

First Name:

6. If necessary, use the Selection tool to reposition the input text field.

Copy a text field

One way to quickly duplicate an object on the Stage is to press Alt while dragging the object. The original object remains in place, allowing you to drag the duplicate. You'll use the Alt key to create two copies of the input text field that you created.

1. On the Stage, with the Selection tool selected, click the input text field that you created and press Alt on your keyboard. Drag a copy of the text field to the right of Last Name.
2. Alt-click the input text field that you just dragged, and then drag the new copy of the field to eMail Address.

First Name:

Last Name:

eMail address:

Assign instance names to text fields

An input text field on the Stage is an instance of the ActionScript TextField object, to which you can apply properties and methods. As a best practice, you should name text field instances so that you or others working on the project can refer to the instance in ActionScript.

1. Select the input text field that you placed next to First Name. In the Property inspector, enter **firstName_txt** in the Instance Name text box.
2. Select the input text field that you placed next to Last Name. In the Property inspector, enter **lastName_txt** in the Instance Name text box.
3. Select the input text field that you placed next to eMail Address. In the Property inspector, enter **eMail_txt** in the Instance Name text box.

To learn more about the TextField object, see “Working with Text” in *Using ActionScript in Flash*.

Create a dynamic text field

Dynamic text can display text from external sources at runtime. Next, you’ll create a dynamic text field that links to an external text file. The external text file you’ll use is named `safetyFeatures.txt`. It is in the same folder as the lesson FLA file. The FLA file already contains ActionScript to display the text when you test or publish the document.

1. In the Timeline, select the Dynamic Text layer. Use the Selection tool to click in the workspace, to deselect any objects.
2. Select the Text tool in the Tools panel.
3. In the text Property inspector, set the following options:
 - Select Dynamic Text from the Text Type pop-up menu.
 - For Text Attributes, set Verdana as the text type, with a font size of 6.
 - For Line Type, select Multiline to ensure the text wraps correctly.
 - Click the text color box and select a dark shade of gray.
 - Select Align Left as a paragraph attribute, if it isn’t already selected.
4. On the Stage, drag to create a text field in the area between the two horizontal rules.



5. In the Instance Name text box of the Property inspector, name the dynamic text field **newFeatures_txt**.

The ActionScript in this document loads an external text file, which is in the same folder as your document. The ActionScript is set up to load the text into a field named `newFeatures_txt`.

Specify format options

The Format Options dialog box allows you to specify margin and indentation settings for the text.

1. With the dynamic text field still selected on the Stage, click Format in the Property inspector.
2. In the Left Margin text box, enter 5, and in the Right Margin text box, enter 5. Then click OK.

The dynamic text now has 5-pixel left and right margins within the text field.

View ActionScript for the dynamic text field

You can view the ActionScript that loads text from the external text file into the dynamic text field. This script uses LoadVars actions to load the safetyfeatures.txt content in the newFeatures text field.

1. In the Timeline, select Frame 1 of the Actions layer.
2. Select Window > Development Panels > Actions, or press F9.

The ActionScript appears as follows:

```
// Load text as variable and assign it to dynamic text field
Features_lv = new LoadVars();
Features_lv.onLoad = onText;
Features_lv.load("safetyfeatures.txt");
function onText() {
    newFeatures_txt.text = Features_lv.safetyFeatures;
}
```

3. Close the Actions panel.

Test the SWF file

Save and test the document to ensure that the dynamic text loads correctly.

1. Select File > Save; then select Control > Test Movie.

In the SWF file window, text from the external text file should appear in the dynamic text field that you created. (If the text does not appear as expected, check that you entered the instance name correctly: newFeatures_txt. Also check that you saved your copy of the practice file in the same folder as the original text_start file.)

2. Type in the input text fields.
3. When you finish testing the file, close the SWF file window.

Check spelling

Flash MX 2004 and Flash MX Professional 2004 introduce new features that let you check spelling in most places where text occurs in your document, including text fields, layer names, and ActionScript strings. To check spelling, you first configure the Spelling Setup options, and then run the spell checker.

1. Select Text > Spelling Setup.
2. Verify that Check Text Fields Content is selected, and that you've selected the appropriate dictionary. Select any other options, as desired. Then click OK.
3. Select Text > Check Spelling and respond to the dialog boxes that the spell checker presents. When you finish checking spelling, save your document.

Note: To check spelling in text in external files, use the spell-check feature associated with the application that you used to create the external file.

Summary

Congratulations on learning the basic ways in which to add text to your document. In a few minutes, you learned how to accomplish the following tasks:

- Create an expanding-width text block
- Create a fixed-width text block
- Edit text and change font attributes
- Select device fonts
- Add an input text field
- Copy a text field
- Assign instance names to text fields
- Create a dynamic text field
- View ActionScript that links the text field to an external text file
- Set up and check spelling

To learn more about the great variety of options you have when adding text to your document, see “Working with Text” in *Using Flash*.

CHAPTER 11

Create a Timeline Animation

Macromedia Flash MX 2004 and Macromedia Flash MX Professional 2004 provide powerful tools for creating animation. Most simple animation in Flash is done using a process known as *tweening*. Tweening is short for “in between” and refers to filling in the frames between two keyframes so that a graphic displayed in the first keyframe changes into the graphic displayed in the second keyframe.

There are two types of tweening you can create in Flash—a *motion tween* and a *shape tween*. The main difference between motion tweening and shape tweening is that motion tweening operates on grouped objects or symbols, and shape tweening is used with objects that are not symbols and are ungrouped.

In this lesson you will learn how to accomplish the following tasks:

- Create an animation using motion tweening
- Create an animation using shape tweening
- Copy and paste keyframes in an animation
- Change the speed of an animation

Set up your workspace

First, you'll open the start file for the lesson and set up your workspace to use an optimal layout for taking lessons.

1. To open your start file, in Flash select File > Open and navigate to the file:
 - On Windows 2000 or XP, browse to *boot drive*\Documents and Settings\All Users\Application Data\Macromedia\Flash MX 2004\language\Configuration\HelpPanel\HowDoI\BasicFlash\start_files and double-click *animation_start fla*.

Note: If the Application Data folder is hidden, you'll need to change your Windows Explorer settings to see the folder.

 - On Windows 98, browse to *boot drive*\Windows\Application Data\Macromedia\Flash MX 2004\language\Configuration\HelpPanel\HowDoI\BasicFlash\start_files and double-click *animation_start fla*.
 - On Macintosh, browse to *Macintosh HD*/Users/Shared/Application Support/Macromedia/Flash MX 2004/language/Configuration/HelpPanel/HowDoI/BasicFlash/start_files and double-click *animation_start fla*.

Note: The BasicFlash\finished_files folder contains completed versions of lesson FLA files, for your reference.
2. Select File > Save As and save the document with a new name, in the same folder, to preserve the original start file.
3. Select Window > Panel Sets > Training Layout to modify your workspace for taking lessons.
4. In the Stage View pop-up menu, in the upper right side of the Timeline, select Show Frame to view both the Stage and the work area.
5. If necessary, drag the lower edge of the Timeline (Window > Timeline) down to enlarge the Timeline view.

You can also use the scroll bar to scroll through the layers.

Create a motion tween

You create a motion tween by defining properties for an instance, a grouped object, or text in a starting keyframe, and then changing the object's properties in a subsequent keyframe. Flash creates the animation from one keyframe to the next in the frames between the keyframes.

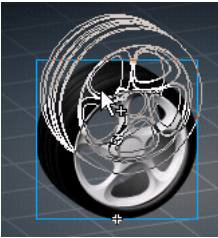
To create a motion tween, you'll take an instance of a tire symbol and make it appear to bounce.

1. In the Timeline (Window > Timeline), double-click the Layer 1 title and type **TireAnim**. Then press Enter or Return to rename the layer.
2. With the TireAnim layer still selected, drag the Tire movie clip from the Library window (Window > Library) to the Stage, positioning it above the tire shadow.

3. Use the Selection tool to reposition the tire, if necessary.



4. With the Selection tool still selected, in the TireAnim layer, select Frame 30. Then press F6 to insert a keyframe.
5. Select Frame 15 and press F6 to add another keyframe.
6. With the playhead still on Frame 15, hold the Shift key to move the tire in a straight line, and drag the tire up.



7. In the TireAnim layer, select any frame between Frames 2 and 14. In the Property inspector, select Motion from the Tween pop-up menu.
An arrow appears in the Timeline between the two keyframes.
8. Select any frame between frames 16 and 29. Again, use the Tween pop-up menu in the Property inspector to select Motion.
9. Select File > Save to save your changes.

Create a shape tween

With shape tweening, you specify attributes for a shape in one keyframe, and then modify the shape or draw another shape in a subsequent keyframe. As with motion tweening, Flash creates the animation in the frames between the keyframes.

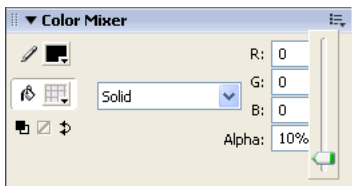
You'll now set up a tween for the tire's shadow so that as the tire bounces, the shadow moves and fades.

1. Click the ShadowAnim layer to select it.
2. Select Frame 30 and press F6 to insert a keyframe; then select Frame 15 and press F6 to insert a keyframe.

3. With the playhead on Frame 15, select the Selection tool. Drag the tire shadow slightly up and to the right.



4. With Frame 15 still selected, select the Eyedropper tool in the Tools panel, and then click the shadow object.
5. If the Color Mixer is not already open, select Window > Design Panels > Color Mixer to open it, then change the Alpha value from 25% to 10%.



6. Click the options menu control in the upper right of the Color Mixer, and select Close Panel to close the panel.
7. Select any frame between Frames 2 and 14 on the ShadowAnim layer. In the Property inspector, select Shape from the Tween pop-up menu.
8. On the ShadowAnim layer, select any frame between Frames 16 and 29. Again, use the Tween pop-up menu in the Property inspector to select Shape.

Copy and paste keyframes in an animation

For the tire to look realistic as it bounces, it should compress slightly on each bounce. You can create this effect by transforming the shape of the tire in Frame 1 of the animation, and copying that frame in Frame 30.

1. With the Selection tool, select Frame 1 of the TireAnim layer. Then press F6 to add a keyframe. A new keyframe is added, and the playhead moves to Frame 2.
2. Select Frame 1 of the TireAnim layer again.
3. In the Tools panel, select the Free Transform tool.
The tire is selected, and transform handles appear around it.



4. Select the transformation center point (the small circle near the center of the movie clip) and drag it to the bottom of the tire.

The center point snaps to the lower middle transform handle.

5. On the Stage, drag the upper middle transform handle down to slightly compress the tire shape.

If necessary, drag the tire to align it over the shadow. To view the positioning, drag the playhead over Frames 1 and 2.

6. Save your file.

Note: Always save your document before attempting to manipulate tweens, including copying, cutting, and pasting frames. If you make an error, you can revert to your saved document.

7. Right-click (Windows) or Control-click (Macintosh) Frame 1 of the TireAnim layer and select Copy Frames from the context menu.

8. Select Frame 29 of the TireAnim layer, and then press F6 to insert a keyframe.

9. In Frame 30 of the TireAnim layer, right-click (Windows) or Control-click (Macintosh) and select Paste Frames from the context menu.

10. Select Control > Test Movie to view the animation.

11. Close the SWF file window to return to the authoring environment.

Change the speed of the animation

When you tested the animation, you might have noticed that the tire bounces rather slowly. You can change the speed of an animation by changing the number of frames that play per second, and by setting positive and negative easing values, which determine the rate of acceleration and deceleration.

Change the frames per second speed

The frame rate, measured in frames per second (fps) is the speed at which the animation plays. By default, Flash animations play at a rate of 12 fps, which is ideal for web animation. Sometimes, however, it's desirable to change the fps rate. You'll now change the frame rate to 36 frames per second, which will make the tire appear to bounce more rapidly.

1. Click on the Stage, away from any objects.
2. In the Property inspector, type **36** in the Frame Rate text box.

The frame rate applies to the entire Flash document, not just to an animation within the document.

Change the acceleration and deceleration

By default, tweened frames play at a constant speed. With easing, you can create a more realistic rate of acceleration and deceleration. Positive values begin the tween rapidly and decelerate the tween toward the end of the animation. Negative values begin the tween slowly and accelerate the tween toward the end of the animation. You'll now add both positive and negative easing values to your animation.

1. On the TireAnim layer, select any frame between Frames 2 and 14. Then in the Property inspector, in the Ease text box, type **100**. Press Enter or Return.
2. On the same layer, select any frame between Frames 16 and 29. Then in the Ease text box, type **-100**. Press Enter or Return.
3. On the ShadowAnim layer, select any frame between Frames 2 and 14. Then in the Ease text box, type **100**. Press Enter or Return.
4. In the same layer, select any frame between Frames 16 and 29. Then in the Ease text box, type **-100**. Press Enter or Return.

Test the SWF file

You'll test your document to view the animation and see if it works as expected.

1. Save your document and select Control > Test Movie.
2. When you finish viewing the animation, close the SWF file window.

Summary

Congratulations on learning how to animate objects in Flash. In just a few minutes, you learned how to accomplish the following tasks:

- Create an animation using motion tweening
- Create an animation using shape tweening
- Copy and paste keyframes in an animation
- Change the speed of an animation

You can also use Timeline effects to quickly add animation effects to text, graphics, images, and symbols. For more information, see “Creating Motion” in *Using Flash*.

To learn more about Flash, take another lesson.

CHAPTER 12

Create a Presentation with Screens (Flash Professional Only)

Flash MX Professional 2004 offers a new way, with slide screens, to create presentations. If you can imagine placing media on slide screens, adding nested slides that inherit media from other slides, and using the built-in controls to navigate through the slides at runtime, then you've imagined exactly how easy it is to create a presentation with slide screens. This lesson demonstrates the simplicity of slides as you accomplish the following tasks:

- Add content to a presentation slide
- Add screen navigation to buttons
- Add and name a slide
- Select and move slides
- Add content to a new slide
- Add transition behaviors to a slide

Set up your workspace

First, you'll open the start file for the lesson and set up your workspace to use an optimal layout for taking lessons.

1. To open your start file, in Flash select File > Open and navigate to the file:
 - On Windows 2000 or XP, browse to *boot drive*\Documents and Settings\All Users\Application Data\Macromedia\Flex MX 2004*language*\Configuration\HelpPanel\HowDoI\BasicFlash\start_files and double-click presentation_start fla.
 - On Windows 98, browse to *boot drive*\Windows\Application Data\Macromedia\Flex MX 2004*language*\Configuration\HelpPanel\HowDoI\BasicFlash\start_files and double-click presentation_start fla.
 - On Macintosh, browse to *Macintosh HD*/Users/Shared/Application Support/Macromedia/Flex MX 2004/*language*/Configuration/HelpPanel/HowDoI/BasicFlash/start_files and double-click presentation_start fla.

Note: The BasicFlash\finished_files folder contains completed versions of lesson FLA files, for your reference.

The document opens in the Flash authoring environment.

2. Select File > Save As and save the document with a new name, in the same folder, to preserve the original start file.

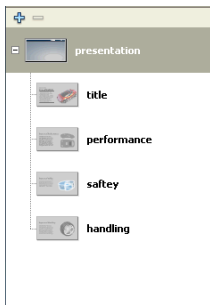
As you complete this lesson, remember to save your work frequently.

3. Select Window > Panel Sets > Training Layout to configure your workspace.

View the screen hierarchy and screen Timelines

You add content to screens in much the same way that you add content to the Stage, but screens are nested movie clips, relying on a hierarchy, nested timelines, and inheritance. All screens exist in the first frame of the root timeline, which is hidden, and all screen content loads on the first frame. For more information about screens, see “Working with Screens (Flash Professional Only)” in *Using Flash*.

1. If the Screen Outline pane is not visible, select Window > Screens.



The Screen Outline pane displays a thumbnail view of each slide in the presentation, and the presentation hierarchy.

When you select a screen in the Screen Outline pane, the screen appears in the Document window. When you select multiple screens, the contents of the first screen selected appear in the Document window.

2. In the Screen Outline pane, select the presentation screen.

All slide screen documents contain a presentation screen, which is at the top level of the screen hierarchy. Think of the presentation slide as a master slide: content on the presentation slide can appear in all slides in your document.

Note: You cannot delete or move the presentation screen.

The four slides that appear indented beneath the presentation slide in the pane represent nested, or child, screens, and the presentation slide is the parent.

3. Open the Timeline, if it's not already open (Window > Timeline). Select another slide in the Screen Outline pane to view the Timeline for that screen.

Each screen has its own Timeline. The main Timeline for a document with screens, however, is never visible.

View screen properties

You can view different properties for a slide, depending on where you select the slide.

1. In the Screen Outline pane, select the presentation thumbnail.

The Property inspector allows you to change the instance name, which is also the name of the screen as it appears in the Screen Outline pane.

2. Select the actual presentation slide, not the thumbnail.

The Property inspector now displays the same controls you're probably used to when working with the Stage and document properties.

Add content to a presentation slide

You'll add navigation buttons to the presentation slide so that the buttons appear on each slide in the presentation.

1. In the Screen Outline pane, select the presentation slide thumbnail. In the Timeline, select Frame 1 of the Navigation layer.
2. From the Library panel, drag the btn, next symbol to the screen, placing it within the black band at the bottom of the screen.
3. In the Property inspector, with the button still selected, enter **280** in the X text box and **165** in the Y text box to place the button.

Note: The coordinates that you enter are in relation to the default center registration point of the slide. For more information about the registration point in screens, see "Specifying the ActionScript class and registration point of a screen (Flash Professional only)" in *Using Flash*.

4. Enter **forwardBtn** in the Instance Name text box.
5. Drag btn, prev to the slide and use the Property inspector to enter **245** in the X text box and **165** in the Y text box.
6. Enter **backBtn** in the Instance Name text box.
7. In the Screen Outline pane, select each nested slide to verify that the buttons now appear on all slides.

Note: Content on a parent screen appears slightly dimmed when you view it on a nested screen.

Add screen navigation behaviors to buttons

When you open a new Flash Slide Presentation, the document already includes functionality that lets users navigate between slide screens using the keyboard arrow keys. You'll also add navigation behaviors to the buttons, offering users an additional option to navigate between slides.

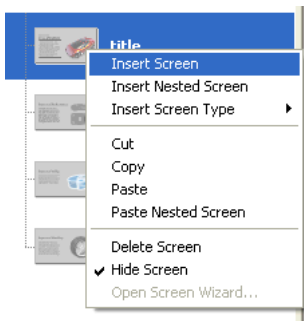
Note: By default, keyboard arrow keys let you navigate between screens on the same level, not between nested screens.

1. On the presentation slide, select the forwardBtn instance. In the Behaviors panel (Window > Development Panels > Behaviors) click the Add (+) button, and then select Screen > Go to Next Slide from the menu.
2. On the presentation slide, select the backBtn instance. In the Behaviors panel (Window > Development Panels > Behaviors) click the Add (+) button, and then select Screen > Go to Previous Slide from the menu.
3. Select Control > Test Movie, and click the buttons in the SWF file window that appears, to ensure your buttons function as expected. When you finish testing your document, close the SWF file window.

Add and name a slide

You can easily add slides to your presentation using the context menu in the Screen Outline pane.

1. In the Screen Outline pane, select the title thumbnail. Right-click (Windows) or Control-click (Macintosh) and select Insert Screen from the context menu.



A new screen appears in the Screen Outline pane, at the same level as the title slide. The new slide automatically inherits media from the presentation slide.

2. Double-click the new slide's name in the Screen Outline pane, and name the slide **features**.

Select and move slides

You can copy, cut, paste, and drag screens in the Screen Outline pane to change their order in the presentation. You'll select three screens, cut them, and paste them so that they're nested as children of the features slide.

1. In the Screen Outline pane, select the performance slide. Press Shift and click the safety and handling slides to add them to the selection.
2. Right-click (Windows) or Control-click (Macintosh) the selected slides and select Cut from the context menu.
3. In the Screen Outline pane, right-click (Windows) or Control-click (Macintosh) the features slide and select Paste Nested Screen from the context menu.

The three slides now appear as children of the features slide.

Add content to a new slide

The features slide, as a child of the presentation slide, inherits properties from that slide. Additionally, because the features slide is a parent to the three slides that you copied and pasted, content that you add to that slide appears on the three children slides.

1. In the Screen Outline pane, select the features thumbnail. From the Library panel, drag the Features Content symbol to anywhere in the Document window.
2. In the Property inspector, give the instance of Features Content an instance name of **features_mc**.
3. In the Property inspector, enter **-275** in the X text box and **-130** in the Y text box to place the instance.

Add transition behaviors

Although your presentation is fairly complete, you'll add transition behaviors to make the presentation more interesting. Specifically, you'll add behaviors that make content fade and appear to fly off the Document window.

1. In the Screen Outline pane, select the features slide. In the Behaviors panel, click the Add (+) button and select Screen > Transition from the menu.
2. In the Transitions dialog box, select Fade from the list of transitions and view the preview in the lower left side of the dialog box. Verify that 2 seconds is selected as the duration and that In is selected as the direction, and click OK.
3. In the Behaviors panel, click Reveal in the Event column to open the pop-up menu and select revealChild. The revealChild option specifies that the behavior will reveal the next child screen.



4. To add the Fly behavior, verify that the features slide is still selected. In the Behaviors panel, click the Add (+) button and select Screen > Transition from the menu.
5. In the Transitions dialog box, select Fly from the list of transitions, and select Out as the direction.
6. In the Duration text box, enter .5 as the length of time to complete the transition.
7. In the Start Location pop-up menu, select Left Center and watch the transition preview, then click OK.
In the Behaviors panel, revealChild now appears twice. With the Fly behavior, however, you want to hide the child screen.
8. In the Behaviors panel, click the second event in the list, which is the one you just added. In the pop-up menu, select hideChild.

Event	Action
revealChild	Transition...
hideChild	Transition...

Test your presentation

Your presentation is now complete and ready to test.

1. Select Control > Test Movie.
2. Use the Forward and Backward navigation buttons to move through the presentation and view the transitions.

Summary

Congratulations on learning how to create a slide presentation with screens. In a few minutes, you learned how to accomplish the following tasks:

- Add content to a presentation slide
- Add screen navigation to buttons
- Add and name a slide
- Select and move slides
- Add content to a new slide
- Add transition behaviors to a slide

To learn more about using screens, see “Working with Screens (Flash Professional Only)” in *Using Flash*.

CHAPTER 13

Add Interactivity with ActionScript

Macromedia Flash MX 2004 and Macromedia Flash MX Professional 2004 offer numerous ways to engage users with interactivity. Additionally, when you incorporate interactivity, you're not limited to playing each Timeline frame sequentially, which offers greater design and development options. You'll learn some of the ways to implement interactivity in this lesson as you accomplish the following tasks:

- Create a new scene
- Write ActionScript to navigate between scenes
- Write ActionScript to play an animated movie clip at runtime
- Use a behavior to play an MP3 file

Note: If you have Macromedia Flash MX Professional 2004, you can use screens to create a document. Screens provide additional navigation options. Slide screens, for example, already include built-in navigation that allows users to move through screens using keyboard arrow keys. For more information about screens, select Help > How Do I > Basic Flash > Create a Presentation with Screens (Flash Professional Only).

Set up your workspace

First, you'll open the start file for the lesson and set up your workspace to use an optimal layout for taking lessons.

1. To open your start file, in Flash select File > Open and navigate to the file:

- On Windows 2000 or XP, browse to *boot drive*\Documents and Settings\All Users\Application Data\Macromedia\Flex MX 2004*language*\Configuration\HelpPanel\HowDoI\BasicActionScript\start_files and double-click *interactivity_start fla*.

Note: If the Application Data folder is hidden, you'll need to change your Windows Explorer settings to see the folder.

- On Windows 98, browse to *boot drive*\Windows\Application Data\Macromedia\Flex MX 2004*language*\Configuration\HelpPanel\HowDoI\BasicActionScript\start_files and double-click *interactivity_start fla*.
- On Macintosh, browse to *Macintosh HD*/Users/Shared/Application Support/Macromedia/Flex MX 2004/*language*/Configuration/HelpPanel/HowDoI/BasicActionScript/start_files and double-click *interactivity_start fla*.

Note: The BasicActionScript\finished_files folder contains completed versions of lesson FLA files, for your reference.

2. To open your start file, in Flash select File > Open. Then do one of the following:

- If you're using a Windows 2000 or XP operating system, browse to *boot drive*\Documents and Settings*username*\Local Settings\Application Data\Macromedia\Flex MX 2004*language*\Configuration\HelpPanel\HowDoI\BasicActionScript\start_files and double-click *interactivity_start fla*.

Note: If the Application Data folder is hidden, you'll need to change your Windows Explorer settings to see the folder.

- If you're using a Windows 98 operating system, browse to *boot drive*\Windows\Application Data\Macromedia\Flex MX 2004*language*\Configuration\HelpPanel\HowDoI\BasicActionScript\start_files and double-click *interactivity_start fla*.
- If you're using a Macintosh operating system, browse to *Macintosh HD*/Users/*username*/Library/Application Support/Macromedia/Flex MX 2004/*language*/Configuration/HelpPanel/HowDoI/BasicActionScript/start_files and double-click *interactivity_start fla*.

Note: The BasicActionScript\finished_files folder contains completed versions of lesson FLA files, for your reference.

The document opens in the Flash authoring environment.

3. Select File > Save As and save the document with a new name, in the same folder, to preserve the original start file.

As you complete this lesson, remember to save your work frequently.

4. Select Window > Panel Sets > Training Layout to configure your workspace.

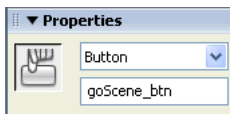
5. In the Stage View pop-up menu, in the upper right of the Timeline, select Show Frame to view both the Stage and the work area.

6. Click in the work area, away from objects on the Stage, so that no objects are selected.

Name button instances

You'll provide instance names for the buttons on the Stage so that you can refer to the instance names in ActionScript.

1. On the lower right side of the Stage, select the button at the left. In the Property inspector (Window > Properties), type **goScene_btn** in the Instance Name text box to name the instance of the symbol.



2. Select the middle button and use the Property inspector to give the button an instance name of **attachMovie_btn**.
3. Select the far right button and use the Property inspector to give the button an instance name of **playSound_btn**.

Add a scene

You can use scenes in Flash to organize your document into discrete sections that can contain content exclusive of other scenes. You'll create and add content to a new scene.

1. Select Insert > Scene.

You can no longer see Scene 1, and Scene 2 now appears above the Stage. The Stage is empty.

2. From the Library panel (Window > Library), drag the Animation movie clip to the Stage. With the movie clip selected, use the Property inspector to give the instance an *x* coordinate of **200** and a *y* coordinate of **15**. Press Enter or Return.

The movie clip moves to the designated Stage coordinates.

3. Use the Property inspector to give the Animation movie clip an instance name of **animation_mc**.
4. Rename Layer 1 **Animation**. Create a new layer and name it **Buttons**. Drag an instance of the **BTNback** symbol to the Stage, and place it anywhere to the right of the movie clip.
5. Use the Property inspector to give the button an instance name of **back_btn**.

Move between scenes

You can move between scenes in the authoring environment by selecting a scene in the Scene panel.

1. To open the Scene panel, select Window > Design Panels > Scene.
2. Select Scene 1.

Control the document with a stop() action

When you test or publish a Flash document that contains more than one scene, by default the scenes play linearly, in the order in which they appear in the Scenes panel. You'll use a `stop()` action for Scene 1 so that the playhead in the Timeline stops at Frame 1 of Scene 1.

1. In the main Timeline for Scene 1, add a new layer and name it **Actions**.
2. Click Frame 1 of the Actions layer. In the Script pane of the Actions panel (Window > Development Panels > Actions), type the following comment followed by the script that stops the playhead on the frame:

```
// stops the playhead at Frame 1
stop();
```

Link a button to a scene

Now that the playhead stops at Frame 1, you'll add ActionScript that takes the user to Scene 2 upon releasing the `goScene_btn` instance.

- Press Enter or Return twice and type the following comment, then write the function that takes users to Scene 2 upon release of the `goScene_btn` instance:

```
// This script takes the user to Scene 2 when goScene_btn is released
goScene_btn.onRelease = function (){
    gotoAndStop("Scene 2", 1);
};
```

In the script that you just typed, you used the `onRelease()` method for the button object. The `gotoAndStop` function is a Timeline control function that lets you specify the scene and frame number. In this case, you specified Frame 1.

Add navigation to return to Scene 1

The function that you'll add to the button in Scene 2, to return the user to Scene 1, is similar to the function that you wrote to take the user to Scene 2.

1. In the Scene panel, select Scene 2. In the Timeline, create a new layer and name it **Actions**.
2. Select Frame 1 of the Actions layer. In the Actions panel, enter the following in the Script pane:

```
// function takes user to Scene 1 when back_btn instance is released.
back_btn.onRelease = function (){
    gotoAndStop("Scene 1", 1);
};
```

The only differences between this function and the function in Scene 1 are the name of the button and the name of the scene.

3. In the Scene panel, select Scene 1.

Play a movie clip

You can configure your document to play a movie clip at runtime. Using the `attachMovie()` method, you can attach an instance of a movie clip in the Library panel to the Stage even though you have not placed an instance on the Stage.

With the `attachMovie()` method, you must export the symbol for ActionScript and assign it a unique linkage identifier, which is different from the instance name.

1. In the Library panel, right-click the MCTrio symbol and select Linkage from the context menu.
2. In the Linkage Properties dialog box, select Export for ActionScript.
3. In the Identifier text box, verify that MCTrio appears as the linkage name.
4. Verify that Export in First Frame is selected, and then click OK.

Movie clips that are exported for use with ActionScript load, by default, before the first frame of the SWF file that contains them. This can create a delay before the first frame plays. When you assign a linkage identifier to an element, you can specify that the movie clip load on the first frame to avoid playback delays.

Use the `attachMovie()` method to play a movie clip

You'll now use the `attachMovie()` method to load the movie clip and provide the symbol with an instance name. Since the instance of the symbol does not exist on the Stage, you'll create the instance name programmatically.

1. In the Timeline, select Frame 1 of the Actions layer for Scene 1.
2. In the Script pane of the Actions panel, place the insertion point at the end of your last line of code. Press Enter or Return, and then type the following to add a comment and create a new function:

```
//function plays trio_mc when attachMovie_btn instance is released.  
attachMovie_btn.onRelease = function(){
```

Next, you'll specify what the function does: it plays the movie clip on the root Timeline, which is the main Timeline. In your script, you'll refer to the movie clip by the linkage identifier name in the Linkage Properties dialog box (MCTrio).

Additionally, even though you didn't place an instance of the MCTrio symbol on the Stage, you'll use ActionScript to create an instance name for the symbol. The instance name that you'll specify is `trio_mc`.

3. With the insertion point at the end of the last line of script, press Enter or Return. Then type the following:

```
_root.attachMovie("MCTrio", "trio_mc", 1);
```

The number 1 in the script that you just typed refers to the depth on the layer in which to play the movie clip.

Every movie clip instance has its own *z* axis (depth) that determines the stacking order of a movie clip within its parent SWF file or movie clip. When you create a new movie clip at runtime using the `attachMovie()` method, you always specify a depth for the new clip as a method parameter.

Note: For more information about the `attachMovie()` method, see `attachMovie()` in the *Flash ActionScript Language Reference*. Additionally, you can use the *Flash ActionScript Language Reference* for information about ActionScript that allows you to manage depth; `getNextHighestDepth()`, `getDepth()`, `getInstanceAtDepth()` are methods of the `MovieClip` class. The `DepthManager` class allows you to manage the relative depth assignments of a movie clip.

Specify movie clip Stage coordinates

In addition to the z axis for the movie clip, you must specify the x and y coordinates to place the movie clip within the Stage area at runtime.

- Press Enter or Return after the last line in the Script pane and type the following:

```
trio_mc._x = 275;
trio_mc._y = 200;
};
```

Unload the movie clip

After the movie clip plays, you need a way to remove the movie clip from the Stage when the user goes to Scene 2. You can modify your script for the `goScene_btn` to “unload” the movie clip.

1. In the Timeline, select Frame 1 of the Actions layer. Then click at the end of the following line of script in the Script pane, within the function that takes the user to Scene 2, to place the insertion point:

```
gotoAndStop("Scene 2", 1);
```

2. Press Enter or Return and type the following script, which unloads the movie clip when the function runs, so that the movie clip does not continue to play when the user goes to Scene 2:

```
unloadMovie("trio_mc");
```

Your entire function for the `goScene_btn` should appear as follows:

```
// This script takes user to Scene 2 when goScene_btn instance is released
goScene_btn.onRelease = function() {
    gotoAndStop("Scene 2", 1);
    unloadMovie("trio_mc");
};
```

Your entire script should appear as follows:

```
// stops the playhead at Frame 1
stop();
```

```
// This script takes user to Scene 2 when goScene_btn instance is released
goScene_btn.onRelease = function() {
    gotoAndStop("Scene 2", 1);
    unloadMovie("trio_mc");
};
```

```
// This function plays trio_mc when attachMovie_btn instance is released.
attachMovie_btn.onRelease = function() {
    _root.attachMovie("MCTrio", "trio_mc", 1);
    trio_mc._x = 275;
    trio_mc._y = 200;
};
```

Use a behavior to play an MP3 file

When you want to add interactivity to your document with ActionScript, you can often rely on behaviors to add the ActionScript for you. You'll use a sound behavior to play an MP3 file from the library.

1. In the Library panel, right-click (Windows) or Control-click (Macintosh) `ping.mp3` and select `Linkage` from the context menu.
2. In the Linkage Properties dialog box, select `Export for ActionScript` and verify that `Export in First Frame` is selected.
3. Verify that `ping.mp3` appears in the Identifier text box, and click `OK`.
4. On the Stage, select the `playSound_btn` instance.
5. In the Behaviors panel (`Window > Development Panels > Behaviors`), click the `Add (+)` button and select `Sound > Load Sound from Library`.
6. In the linkage ID text box, enter `ping.mp3`, and in the Name text box below, enter `ping`. Then click `OK`.

Test your document

Test your document to verify that the interactivity works as expected.

1. Select `Control > Test Movie`.
2. In the SWF file, click the left button to play Scene 2. When you finish viewing Scene 2, click the `Back` button.
3. Click the middle button in Scene 1 to see the movie clip play.
4. Click the right button to play the MP3 sound.
5. Click the left button again to verify that the movie clip unloads.

Summary

Congratulations on learning how to create an interactive document. In a few minutes, you learned how to accomplish the following tasks:

- Create a new scene
- Write ActionScript to navigate between scenes
- Write ActionScript to play an animated movie clip at runtime
- Use a behavior to play an MP3 file

To learn more about ActionScript, take another lesson in the Basic ActionScript series.

CHAPTER 14

Create a Form with Conditional Logic and Send Data

You can create a form with conditional logic that allows the SWF file to respond in different ways, depending on user interaction, and send the form data from the SWF file to an external source. In this lesson, you will create a simple form as you complete the following tasks:

- Add an input text field to collect form data
- Create a button symbol
- Add a `stop()` action
- Write a script that validates the form with conditional logic
- Pass data out of a SWF file
- Write a function

Before taking this lesson, you should be familiar with writing functions and variables; to learn about these, select Help > How Do I > Quick Tasks > Create an Application.

Set up your workspace

First, you'll open the start file for the lesson and set up your workspace to use an optimal layout for taking lessons.

1. To open your start file, in Flash select File > Open and navigate to the file:
 - On Windows 2000 or XP, browse to *boot drive*\Documents and Settings\All Users\Application Data\Macromedia\Flex MX 2004*language*\Configuration\HelpPanel\HowDoI\BasicActionScript\start_files and double-click `simpleForm_start fla`.

Note: If the Application Data folder is hidden, you'll need to change your Windows Explorer settings to see the folder.

 - On Windows 98, browse to *boot drive*\Windows\Application Data\Macromedia\Flex MX 2004*language*\Configuration\HelpPanel\ HowDoI\BasicActionScript\start_files and double-click `simpleForm_start fla`.
 - On Macintosh, browse to *Macintosh HD*/Users/Shared/Application Support/Macromedia/Flex MX 2004/*language*/Configuration/HelpPanel/HowDoI/BasicActionScript/start_files and double-click `simpleForm_start fla`.

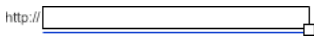
Note: The BasicActionScript\finished_files folder contains completed versions of lesson FLA files, for your reference.

2. Select File > Save As and save the document with a new name, in the same folder, to preserve the original start file.
3. Select Window > Panel Sets > Training Layout to modify your workspace for taking lessons.
4. If necessary, drag the lower edge of the Timeline (Window > Timeline) down to enlarge the Timeline view.

Add an input text field to collect form data

You'll start by inserting a text field in your document in which viewers can enter data. You will also name the text field so you can later reference the text field as you use ActionScript in your Flash application.

1. Click in the work area, away from objects on the Stage, so that no objects are selected.
2. In the Tools panel, select the Text tool.
3. In the Property inspector, do the following to set text attributes:
 - Select Input Text from the Text Type pop-up menu.
 - Select `_sans` from the Font pop-up menu.
 - Enter **10** in the Font Size text box.
 - Click the text color box and select a shade of dark blue.
 - Verify that Align Left is selected.
 - Verify that Single Line is selected in the Line Type pop-up menu.
4. In the Timeline, select Frame 1 of the Input Text layer.
5. On the Stage, drag the Text tool to create an input text field to the right of the `http://` text.



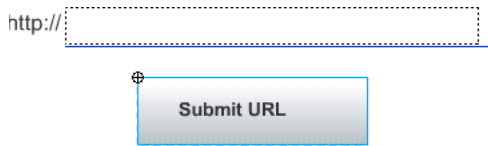
6. If necessary, use the Selection tool to drag the text field or use the arrows keys to adjust the position.
7. With the input text field still selected, in the Property inspector, type `url_txt` in the Instance Name text box.

You'll refer to the instance name later when you add ActionScript.

Add a Submit button to the form

The Library panel contains a Submit button symbol that you will add to the form.

1. From the Library panel (Window > Library), drag the Submit button to the Stage and place it over the SubmitURL guide.

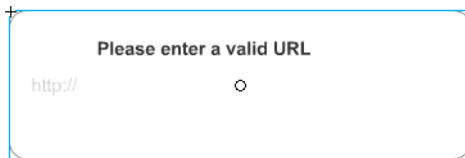


2. Drag the button or use the arrows keys to adjust the position, if necessary.
3. In the Property inspector, type `submit_btn` in the Instance Name text box.

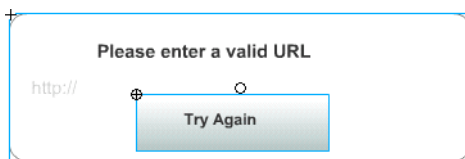
Add an error message

You'll add a message to display if the user clicks the Submit button before entering data.

1. In the Timeline, with the Branding layer selected, click the Insert Layer button and name the layer **Dialog Boxes**.
2. Select Frame 5 of the Dialog Boxes layer. Right-click (Windows) or Control-click (Macintosh) the selected layer and select Insert Blank Keyframe from the context menu.
3. From the Library panel, drag the Dialog Box-error movie clip to the center of the Stage.



4. In the Timeline, select Frame 5 of the Buttons layer. Drag the Try Again Button symbol from the Library panel to the Stage, placing the button below the error message text.



5. With the button still selected, in the Property inspector, enter `tryAgain_btn` in the Instance Name text box.

Add a confirmation message

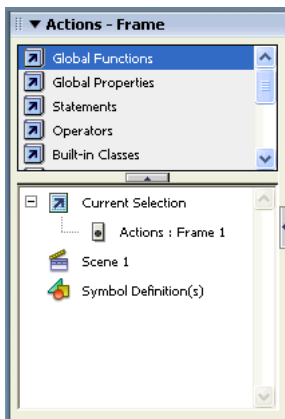
Next, you'll add a message to display when the user submits an entry in the text field.

1. In the Dialog Boxes layer, select Frame 10. Right-click (Windows) or Control-click (Macintosh) the selected frame and select Insert Blank Keyframe from the context menu.
2. From the Library panel, drag the Dialog Box-confirm movie clip to the center of the Stage.
3. Close the Library panel.

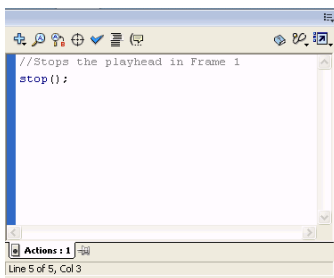
Add a stop() action

When a Flash application loads in Flash Player, it automatically plays in a continuous loop. You use ActionScript to control playhead movement in a Timeline. You will add a `stop()` action in the first frame of your application so the user can make an entry in the input text field.

1. In the Timeline, with the Input Text layer selected, add a new layer and name it **Actions**.
2. Select Frame 1 of the Actions layer.
3. In the Actions panel (Window > Development Panels > Actions), verify that Frame 1 is selected.



4. Click in the Script pane of the Actions panel, and type the following comment: `//Stops the playhead at Frame 1.` Press Enter or Return.
5. Type `stop();` to add the stop action.



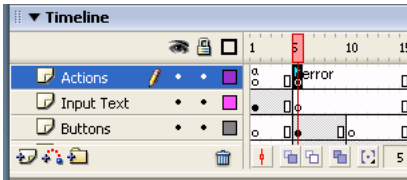
Add frame labels for navigation

When the viewer presses the Submit button, you want Flash to jump to either the error message or the confirmation message, depending on what is entered in the text field. Labeling a frame helps you easily reference it in ActionScript. This is helpful for sending the playhead to a specific frame.

Next, you'll add frame labels to help you navigate your Flash application.

1. Add a keyframe (Insert > Timeline > Keyframe) to Frame 5 of the Actions layer.
2. In the Property inspector, type **error** in the Frame Label text box. Press Enter or Return.

The label text and a flag appear in Frame 5 of the Timeline.



3. Add a keyframe to Frame 10 of the Actions layer. In the Property inspector, type **confirm** in the Frame Label text box. Press Enter or Return.

Add conditional logic for the Submit button

With ActionScript, you can have Flash compare information and take action based on criteria you specify. In this example, you'll add ActionScript for Flash to take one action if the user enters no data in the text field, and a different action if the user does.

1. Select Frame 1 of the Actions layer. In the Script pane, place the insertion point after the `stop();` code. Press Enter or Return.
2. Type the following comment: `//Adds conditional logic for the Submit button that validates user input.` Press Enter or Return.
3. In the Actions panel, click the Insert a Target Path button, located at the top of the panel.
4. In the Insert Target Path dialog box, verify that Relative is selected. Click `submit_btn` on the hierarchical tree, and click OK.
5. In the Script pane, type a period (`.`) after `submit_btn`, and then type `onRelease`.
6. With the insertion point after `onRelease`, type `= function (){}` in the Script pane.
7. Place the insertion point between the curly braces and press Enter or Return, and then type `if (url_txt.text == null || url_txt.text == ""){` in the Script pane.
The parallel lines are equivalent to logical *or* in ActionScript.
8. With the insertion point still inside the curly braces, press Enter or Return.
9. Type `gotoAndStop("error");` in the Script pane. Press Enter or Return.
10. Place the insertion point after the curly brace and type `else{` in the Script pane. Press Enter or Return.

11. Type `gotoAndStop("confirm")` in the Script pane. Press Enter or Return, and type `},` then press Enter or Return again and type `};`.

Your script should appear as follows:

```
//Stops the playhead at frame 1
stop();
//Adds conditional logic for the Submit button that validates user input
this.submit_btn.onRelease = function(){
    if (url_txt.text == null || url_txt.text == ""){
        gotoAndStop("error");
    } else {
        gotoAndStop("confirm")
    }
};
```

Pass data out of a SWF file

You can send data from a Flash application in various ways—for example, in this lesson you send data to a web server to load a web page in the browser. After the `else` statement, you'll add the ActionScript to have Flash go to the URL the viewer enters in the input text field.

1. In the Script pane, place the insertion point in front of the line that reads `gotoAndStop("confirm").`
2. Then in the Actions toolbox, select **Global Functions > Browser/Network** and double-click `getUrl`.
3. With the insertion point within the `getUrl()` parentheses, type `"http://"+url_txt.text` to specify the data that should pass from the SWF file. (Do not leave spaces in the code.)

Your script should look like the following:

```
stop();
this.submit_btn.onRelease = function(){
    if (url_txt.text == null || url_txt.text == ""){
        gotoAndStop("Error");
    } else {
        getUrl ("http://"+url_txt.text);
        gotoAndStop("Confirm")
    }
};
```

Write a function for the Try Again button

A function is a script that you can use repeatedly to perform a specific task. You can pass parameters to a function, and it can return a value. In this example, when the user clicks the Try Again button, a function runs that returns the playhead to Frame 1.

You'll write that function now. In this script, you'll type the frame number, because we did not label Frame 1.

1. In the Timeline, select Frame 5 of the Actions layer.
2. In the Script pane, type the following comment, `//button function returns user to Frame 1.` Press Enter or Return.

3. Type `tryAgain_btn.onRelease = function(){`, then press Enter or Return.
4. Type `gotoAndStop(1);`, then press Enter or Return and type `}` to complete the script.

Test your SWF file

You'll test your document by entering a URL and checking whether it works as expected.

1. Select Control > Test Movie.
2. When the SWF file appears, click the Submit button before typing anything in the input text field.

The error message appears.

3. Click the Try Again button, and then type the URL of a valid website in the input text field. Click the Submit button.

Your default browser opens the web page.

Summary

Congratulations on learning how to write a script with conditional logic and send data. In a few minutes, you learned how to complete the following tasks:

- Add an input text field to a document
- Create a button symbol
- Add a `stop()` action
- Write a script that validates the form with conditional logic
- Pass data out of a SWF file
- Write a function

To learn more about Flash, take another lesson.

CHAPTER 15

Work with Objects and Classes Using ActionScript 2.0

Classes are the blueprint for objects in Macromedia Flash MX 2004 and Macromedia Flash MX Professional 2004. All objects in Flash have an underlying class; for example, all movie clips have a method called `getURL`, and `getURL` is defined in the class definition for a movie clip. Flash contains many predefined classes, such as the `MovieClip` class, the `Array` class, the `Color` class, and the `CheckBox` class.

In this lesson, you'll learn about creating and modifying classes. Specifically, you'll learn how to accomplish the following tasks:

- Create objects from existing classes
- Create a custom class
- Use datatyping with custom classes
- Extend an existing class

Note: This lesson is designed for Flash developers who are familiar with basic Flash and ActionScript concepts.

Set up your workspace

First, you'll view the finished files and set up your workspace to use an optimal layout for taking lessons.

1. View the finished files. This lesson does not include start files. You can find finished files of `handson1.fla`, `handson2.fla`, `handson3.fla`, `Product.as`, and `Drag.as`, which are examples of the files that you'll create in this lesson:

- On Windows 2000 or XP, browse to *boot drive*\Documents and Settings\All Users\Application Data\Macromedia\Flex MX 2004\language\Configuration\HelpPanel\HowDoI\BasicActionScript\finished_files.

Note: If the Application Data folder is hidden, you'll need to change your Windows Explorer settings to see the folder.

- On Windows 98, browse to *boot drive*\Windows\Application Data\Macromedia\Flex MX 2004\language\Configuration\HelpPanel\HowDoI\BasicActionScript\finished_files.

- On Macintosh, browse to *Macintosh HD/Users/Shared/Application Support/Macromedia/Flash MX 2004/language/Configuration/HelpPanel/HowDoI/BasicActionScript/finished_files*.
2. Select Window > Panel Sets > Training Layout to configure your workspace.

About classes and object types

A *class*, also referred to as an *object type*, is like a blueprint. An object doesn't exist until you create it, or instantiate it, from a class definition. An object is an instance of a class.

Properties are the characteristics of an object. For example, when you align movie clips, you change the `_x` and `_y` properties of the `MovieClip` object. A property is a variable that is attached to a class. A property can either be public, which means it is accessible outside the class, or private, which means it can be accessed only within the class.

In object-oriented terms, a method is a behavior or procedure that can act on the object. A hypothetical `throw()` method on a ball would know the size and weight of the ball. A method is aware of the object and all the properties that it contains and can work on that object.

Creating an object from a class

You'll create an existing class using visual tools (the `TextField` class) and code (using the `Date` class).

1. Open a new Flash document and change the name of Layer 1 to **Text**.
2. In the Text layer, create a dynamic text field and assign it the instance name of **currentDate_txt**.
3. Create an Actions layer. With Frame 1 of the Actions layer selected, open the Actions panel.
4. Create, or instantiate, an object from the `Date` class, named **myDate**:

```
var myDate:Date=new Date();
```

5. Create a variable called `currentMonth` equal to the `getMonth()` method:

```
var currentMonth:Number = myDate.getMonth();
```

6. Trace the value of `currentMonth`:

```
trace (currentMonth);
```

7. Save and test the document.

You should see a number in the Output panel that represents the month.

The `getMonth()` method displays the current month. The `getMonth()` method is zero-indexed, meaning the numbering begins at zero rather than one, so the number displayed is one less than what you would expect.

8. Close the Output panel and the SWF file window.

Modify your script

You'll modify your script to compensate for the zero indexing.

1. Add **+1** to the value when you create `currentMonth`, and test your document to be sure the expected month number appears. That line of script should read as follows:

```
var currentMonth:Number = myDate.getMonth()+1;
```

2. Comment the trace statement:

```
//trace (currentMonth);
```

3. Below the trace statement, set the `autoSize` property of your text box to `true`:

```
currentDate_txt.autoSize = true;
```

4. Use the `text` property of your text box to display today's date in the form Today is mm/dd/yyyy. Use the `currentMonth` variable you have already created, plus the `getDate()` and `getFullYear()` methods of the `Date` object:

```
currentDate_txt.text="Today is "+currentMonth+"/"+ myDate.getDate() + "/"  
"+myDate.getFullYear();
```

5. Verify that your script appears as follows:

```
var myDate>Date=new Date();  
var currentMonth:Number = myDate.getMonth()+1;  
//trace (currentMonth);  
currentDate_txt.autoSize = true;  
currentDate_txt.text="Today is "+currentMonth+"/"+ myDate.getDate() + "/"  
"+myDate.getFullYear();
```

6. Save and test the document. The current date should appear in the SWF file window.

Note: An example finished file of the document you just created, named `handson1 fla`, is located in your finished files folder. For the path, see ["Set up your workspace" on page 113](#).

Creating a custom class

While ActionScript includes many classes of objects, such as the `MovieClip` class and the `Color` class, there will be times when you need to construct your own classes so you can create objects based on a particular set of properties or methods.

To create a class that defines each of the new objects, you create a constructor for a custom object class and then create new object instances based on that new class, as in the following example:

Note: The following ActionScript is an example only. You should not enter the script in your lesson FLA file.

```
function Product (id:Number, prodName:String, price:Number)  
{  
    this.id:Number = id;  
    this.prodName:String = prodName;  
    this.price:Number = price;  
}
```

In order to properly define a class in ActionScript 2.0, you must surround all classes by the `class` keyword, and you must declare all variables in the constructor outside of the constructor.

Following is an example:

Note: The following ActionScript is an example only. You should not enter the script in your lesson FLA file.

```
class Product
{
//variable declaration
var id:Number
var productName:String
var price:Number
//constructor
function Product (id:Number, prodName:String, price:Number)
{
this.id = id;
this.prodName = prodName;
this.price = price;
}
}
```

To create objects from this class, you could now use the following code:

Note: The following ActionScript is an example only. You should not enter the script in your lesson FLA file.

```
var cliplessPedal:Product=new Product(1, "Clipless Pedal", 11);
var monkeyBar:Product=new Product(2, "Monkey Bar", 10);
```

However, in ActionScript 2.0, variables that are part of a class structure should not be accessed directly. You should write methods within the class that will access these variables directly. There should be different methods that get and set properties (known as “getter” and “setter” methods). You must indicate the data type for both a method’s return value and any parameters that are passed to the method when it is called.

Specify the data type for method return values

You must indicate data types for values returned by methods after the method name and list of parameters, as in the following example:

Note: The following ActionScript is an example only. You should not enter the script in your lesson FLA file.

```
public function getProductName() :String
{
return name;
}
```

If no value is returned (for example, a property is being set), the data type is `Void`:

```
public function setProductName(productName:String) :Void
{
this.productName=productName;
}
```

Build a custom class

You'll now build a new Product class with getter and setter methods and create an object from the Product class.

1. Create an ActionScript file by doing one of the following:
 - If you're using Flash MX 2004 Professional, select File > New > ActionScript File (Not Flash Document). Save the document with the name Product.
 - If you're using Flash MX 2004, open a text editor, such as Notepad. Save the file with the name Product.as. (Remember to give the file the AS extension, to create an ActionScript file.)

2. Create a constructor for a Product class by creating a function called Product that takes the arguments id, prodName, and description:

```
function Product (id:Number, prodName:String, description:String)
{ }
```

3. In the constructor function, set the properties of the Product class equal to the setter methods that you will create:

```
setID(id);
setProdName(prodName);
setDescription(description);
```

4. Wrap the constructor function around the class keyword. Be sure to declare each variable used in the class:

```
class Product
{
    var id:Number;
    var prodName:String;
    var description:String

    function Product (id:Number, prodName:String, description:String)
    {
        setID(id);
        setProdName(prodName);
        setDescription(description);
    }
}
```

5. Define getter and setter methods for each property of the class, as in the following example. Be sure to specify Void as the return type for the setter methods, and indicate the data type returned for the getter methods.

```
class Product
{
    var id:Number;
    var prodName:String;
    var description:String

    function Product (id:Number, prodName:String, description:String) {
        setID(id);
        setProdName(prodName);
        setDescription(description);
```

```

}
    public function setID (id:Number) :Void
    {
        this.id = id;
    }
    public function setProdName (prodName:String) :Void
    {
        this.prodName = prodName;
    }
    public function setDescription (description:String) :Void
    {
        this.description = description;
    }
    public function getID () :Number {
        return id;
    }
    public function getProdName () :String {
        return prodName
    }
    public function getDescription () :String {
        return description;
    }
}
}

```

6. Save your file.

Note: An example finished file of the file you just created, named Product.as, is located in your finished files folder. For the path, see [“Set up your workspace” on page 113](#).

Create two objects from the Product class

You'll create a new FLA file, then create two objects from the Product class.

1. Open a new Flash document and save it in the same location where you saved Product.as.
2. In the new document, select Frame 1 in the Timeline. In the Actions panel, create two objects from the Product class using the data shown in the following table (the ActionScript that you'll create appears after the table).

Instance name	Data	
pedals	id	0
	prodName	Clipless Pedals
	description	Excellent cleat engagement
handleBars	id	1
	prodName	ATB
	description	Available in comfort and aero design

3. Verify that you created the objects as follows:

```
var handleBars:Product = new Product (1, "ATB", "Available in comfort and
aero design");
var pedals:Product=new Product(0,"Clipless Pedals","Excellent cleat
engagement");
```

4. Trace the description property of pedals:

```
trace (pedals.getDescription ());
```

5. Save and test the document. You should see the description of pedals in the Output panel.

Note: An example finished file of the document you just created, named handson2.fla, is located in your finished files folder. For the path, see [“Set up your workspace” on page 113](#).

About extending existing classes

The `extends` keyword in ActionScript 2.0 allows you to use all the methods and properties of an existing class in a new class. For example, if you wanted to define a class that inherited everything from the `MovieClip` class, you could use the following:

```
class Drag extends MovieClip
{ }
```

The `Drag` class now inherits all properties and methods from the existing `MovieClip` class, and you can use `MovieClip` properties and methods anywhere within the class definition, as in the following example:

Note: The following ActionScript is an example only. You should not enter the script in your lesson FLA file.

```
class Drag extends MovieClip
{
    // constructor
    function Drag ()
    {
        onPress=doDrag;
        onRelease=doDrop;
    }
    private function doDrag():Void
    {
        this.startDrag();
    }
    private function doDrop():Void
    {
        this.stopDrag();
    }
}
```

Note: The Convert to Symbol dialog box now offers a class field in which you can associate visual objects (such as movie clip) with any class that you define in ActionScript 2.0.

Extend the MovieClip class to create a new class

You'll create a new class by extending the built-in MovieClip class.

1. Create a new Flash document and name it Shape.fla.
2. Using the drawing tools, draw a shape on the Stage. With the entire shape selected, right-click (Windows) or Control-click (Macintosh) the shape and select Convert to Symbol from the context menu.
3. In the Convert to Symbol dialog box, select Movie Clip as the behavior, and click Advanced. Select Export for ActionScript.
4. In the Name text box, enter **myShape**.
5. In the AS 2.0 Class text box, enter **Drag**. Click OK.

This associates the movie clip with the Drag class that you'll create.

6. Using the Property inspector, assign the movie clip an instance name. Then save the FLA file.

Note: An example finished file of the document you just created, named handson3.fla, is located in your finished files folder. For the path, see [“Set up your workspace” on page 113](#).

7. Create an ActionScript file by doing one of the following:
 - If you're using Flash MX 2004 Professional, select File > New > ActionScript File (Not Flash Document). Save the document with the name Drag, in the same location where you saved Shape.fla.
 - If you're using Flash MX 2004, open a text editor, such as Notepad. Save the file with the name Drag.as, in the same location where you saved Shape.fla.
8. In the ActionScript file that you just created, create a new class and constructor called Drag:

```
class Drag extends MovieClip
{
    function Drag ()
    {
        onPress=doDrag;
        onRelease=doDrop;
    }
}
```


9. Define private methods in the class that use the existing movie clip methods, `startDrag()` and `stopDrag()`:

```
class Drag extends MovieClip
{
function Drag()
{
    onPress=doDrag;
    onRelease=doDrop;
}
private function doDrag():Void
{
this.startDrag();
}
private function doDrop():Void
{
    this.stopDrag()
}
}
```

10. Save the ActionScript file.

11. Test the Shape.fla document. You should be able to drag the movie clip.

Note: An example of the ActionScript file you just created, named Drag.as, is located in your finished files folder. For the path, see [“Set up your workspace” on page 113](#).

Summary

Congratulations on learning how to work with objects and classes in ActionScript 2.0. In a few minutes, you learned how to accomplish the following tasks:

- Create and use objects from existing classes
- Create a custom class
- Create a property within a custom class
- Create a method within a custom class
- Extend an existing class and take advantage of inheritance

