

Step-by-Step Tasks in Full Color

Includes

- Workshops ▶
- ACE Exam Objectives ▶
- More than 600 Essential
Flash Professional CS5 ▶
Tasks

Adobe®

Flash Professional CS5

1 2 3 4 5 6 7 **on Demand**

What you need, when you need it!

que®

Steve Johnson, Perspection, Inc.

Adobe® Flash® Professional CS5 On Demand

Copyright © 2010 by Perspection, Inc.

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

Library of Congress Cataloging-in-Publication Data is on file

ISBN-13: 978-0-7897-4443-2

ISBN-10: 0-7897-4443-0

Printed and bound in the United States of America

First Printing: May 2010

13 12 11 10 4 3 2 1

Que Publishing offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales.

For information, please contact: U.S. Corporate and Government Sales

1-800-382-3419 or corpsales@pearsontechgroup.com

For sales outside the U.S., please contact: International Sales

1-317-428-3341 or International@pearsontechgroup.com

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Que cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Adobe, the Adobe logo, Acrobat, Bridge, Device Central, Dreamweaver, Extension Manager, Flash, InDesign, Illustrator, PageMaker, Photoshop, Photoshop Elements, and Version Cue are registered trademarks of Adobe System Incorporated. Apple, Mac OS, and Macintosh are trademarks of Apple Computer, Inc. Microsoft and the Microsoft Office logo are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The authors and the publishers shall have neither liability nor responsibility to any person or entity with respect to any loss or damage arising from the information contained in this book.

Publisher

Paul Boger

Associate Publisher

Greg Wiegand

Acquisitions Editor

Laura Norman

Managing Editor

Steve Johnson

Author

Steve Johnson

Contributor

Matthew David

Technical Editor

Matthew David

Page Layout

Beth Teyler

James Teyler

Interior Designers

Steve Johnson

Marian Hartsough

Photographs

Tracy Teyler

Indexer

Katherine Stimson

Proofreader

Beth Teyler

Team Coordinator

Cindy Teeters

Introduction

Welcome to *Adobe Flash Professional CS5 On Demand*, a visual quick reference book that shows you how to work efficiently with Flash. This book provides complete coverage of basic to advanced Flash skills.

How This Book Works

You don't have to read this book in any particular order. We've designed the book so that you can jump in, get the information you need, and jump out. However, the book does follow a logical progression from simple tasks to more complex ones. Each task is presented on no more than two facing pages, which lets you focus on a single task without having to turn the page. To find the information that you need, just look up the task in the table of contents or index, and turn to the page listed. Read the task introduction, follow the step-by-step instructions in the left column along with screen illustrations in the right column, and you're done.

What's New

If you're searching for what's new in Flash CS5, just look for the icon: **New!**. The new icon appears in the table of contents and throughout this book so you can quickly and easily identify a new or improved feature in Flash. A complete description of each new feature appears in the New Features guide in the back of this book.

Keyboard Shortcuts

Most menu commands have a keyboard equivalent, such as Ctrl+P (Win) or +P (Mac), as a quicker alternative to using the mouse. A complete list of keyboard shortcuts is available on the Web at www.perspection.com.



How You'll Learn

How This Book Works

What's New

Keyboard Shortcuts

Step-by-Step Instructions

Real World Examples

Workshops

Adobe Certified Expert

Get More on the Web



Step-by-Step Instructions

This book provides concise step-by-step instructions that show you “how” to accomplish a task. Each set of instructions includes illustrations that directly correspond to the easy-to-read steps. Also included in the text are time-savers, tables, and sidebars to help you work more efficiently or to teach you more in-depth information. A “Did You Know?” provides tips and techniques to help you work smarter, while a “See Also” leads you to other parts of the book containing related information about the task.

Changing Symbol and Instance Type

Symbols have three specific types that are defined as you first create the symbol, graphic, movie clip, and button. You can change these types for a symbol's instance by selecting it and then selecting a new type option in the Property Inspector. In this video you need the instance to display a behavior different than the parent symbol. Alternatively, you can change the behavior of the parent symbol in the Library. All subsequent instances that you create from this symbol will exhibit the new behavior.

Change a Symbol's Behavior

- 1 Open or expand the Library panel.
- 2 Select the symbol from the list.
- 3 Click the Properties button at the bottom of the Library window to open the Symbol Properties dialog box.
- 4 Select a different symbol type option.
- 5 Click OK.

The symbol displays a new behavior. All subsequent instances will default to this behavior.

Did You Know?

If you are already using instances of a symbol in your movie clip, they will not be affected by a change in behavior. Flash allows you to change the types of instances separately from your symbol's default behavior. To update the instances to a new symbol, select them on the Stage, and then change the type in the Property Inspector.

See Also

See "Resizing Symbols in Part 1" on page 155 for information on cloning and resizing.

194 Chapter 4

Easy-to-follow introductions focus on a single concept.

Illustrations match the numbered steps.

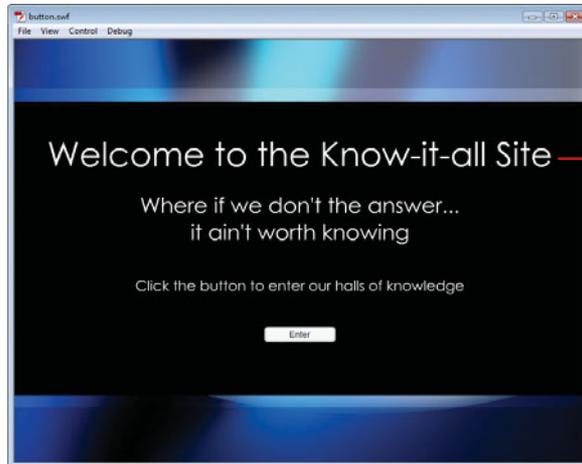
Numbered steps guide you through each task.

See Also points you to related information in the book.

Did You Know? alerts you to tips, techniques and related information.

Real World Examples

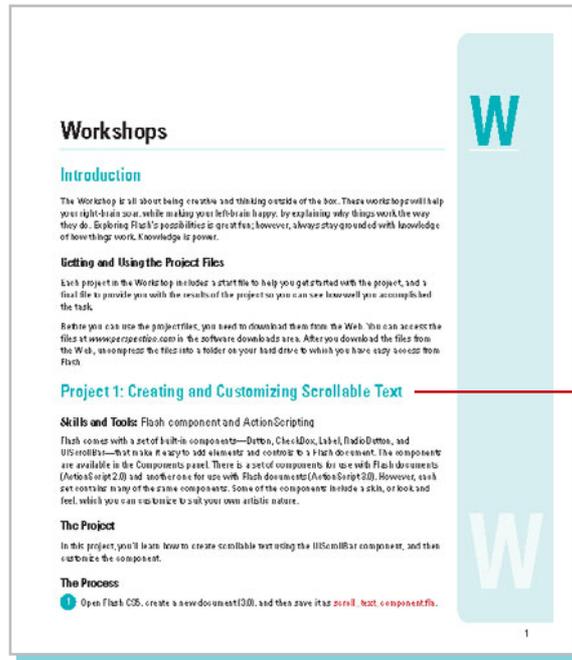
This book uses real world example files to give you a context in which to use the task. By using the example files, you won't waste time looking for or creating sample files. You get a start file and a result file, so you can compare your work. Not every topic needs an example file, such as changing options, so we provide a complete list of the example files used throughout the book. The example files that you need for project tasks along with a complete file list are available on the Web at www.perspection.com.



Real world examples help you apply what you've learned to other tasks.

Workshops

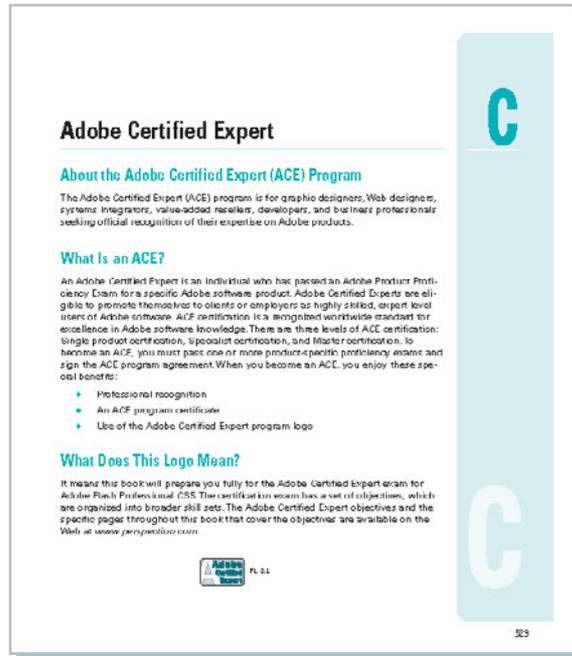
This book shows you how to put together the individual step-by-step tasks into in-depth projects with the Workshop. You start each project with a sample file, work through the steps, and then compare your results with a project results file at the end. The Workshop projects and associated files are available on the Web at www.perspection.com.



The Workshops walk you through in-depth projects to help you put Flash to work.

Adobe Certified Expert

This book prepares you fully for the Adobe Certified Expert (ACE) exam for Adobe Flash CS5. Each Adobe Certified Expert certification level has a set of objectives, which are organized into broader skill sets. To prepare for the certification exam, you should review and perform each task identified with an ACE objective to confirm that you can meet the requirements for the exam. Information about the ACE program is available in the back of this book. The Adobe Certified Expert objectives and the specific pages that cover them are available on the Web at www.perspection.com.



Get More on the Web

In addition to the information in this book, you can also get more information on the Web to help you get up-to-speed faster with Flash CS5. Some of the information includes:

Transition Helpers

- ◆ **Only New Features.** Download and print the new feature tasks as a quick and easy guide.

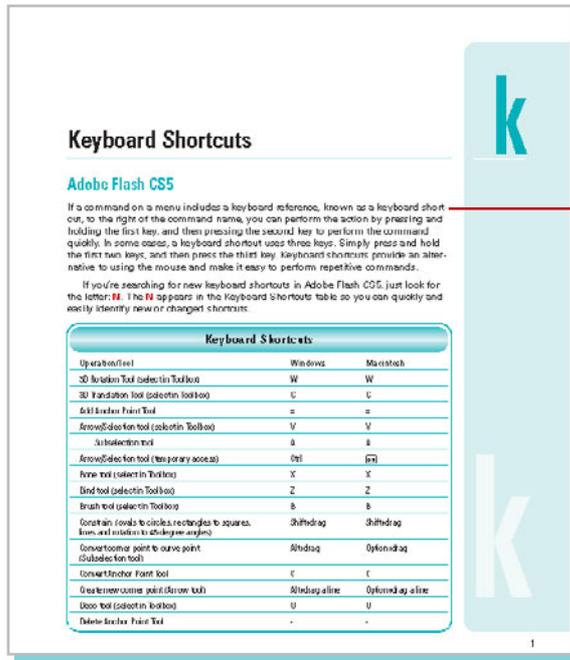
Productivity Tools

- ◆ **Keyboard Shortcuts.** Download a list of keyboard shortcuts to learn faster ways to get the job done.

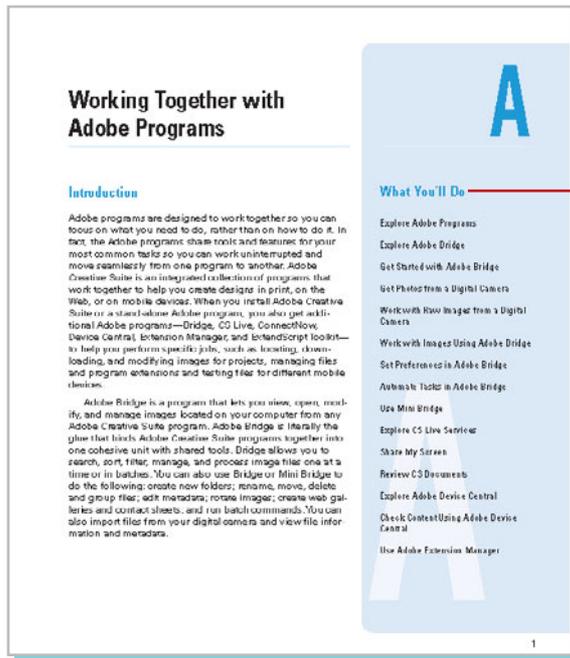
More Content

- ◆ **Photographs.** Download photographs and other graphics to use in your Flash documents.
- ◆ **More Content.** Download new content developed after publication.

You can access these additional resources on the Web at www.queondemand.com or www.perspection.com.



Additional content is available on the Web.



Additional content is available on the Web.

Developing iPhone Apps

Introduction

Apple has this device called an iPhone. Have you heard of it? Of course you have. There is even a good chance you have an iPhone. While the iPhone itself is an amazing piece of technology, it is the software that makes the iPhone really great.

The challenge in developing applications for the iPhone is that you need to know how to develop using Apple's Cocoa Touch framework and the programming language Objective C. One of the most interesting and talked about features in Flash CS5 is Adobe's inclusion of a tool that converts your Flash movies to files that will run on any iPhone OS device without needing to know Objective C. You just need to apply the Flash and ActionScript knowledge you already have. A new tool in Flash CS5, called the iPhone Packager, converts files into valid iPhone applications.

Before you can create an iPhone App there are several activities you need to complete so you can develop in Flash CS5. They are (1) Become a Certified Apple Developer, (2) Create a Development and Developer P12 certificate, and (3) Register your development iPhone with Apple.

In this chapter, you'll get an overview for how to start developing for the iPhone using Flash CS5 including deploying an App to the iTunes App Store.

What You'll Do

Become an Apple Certified Developer

Register an iPhone and other Apple Testing Devices

Create App's ID

Create a Developer's Provisioning Profiles

Create and Publish an iPhone App

Control the Accelerometer

Save Images to the Camera Roll

Understand the Limitations of Flash

Identify Devices to Deploy an Ad Hoc App

Create an Ad Hoc Distribution Profile

Package an Ad Hoc App

Use iTunes Connect to Publish an App

Use Screen Orientation in an App

Use Geolocation in an App

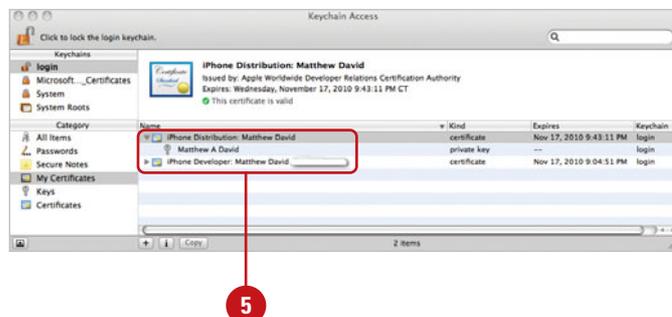
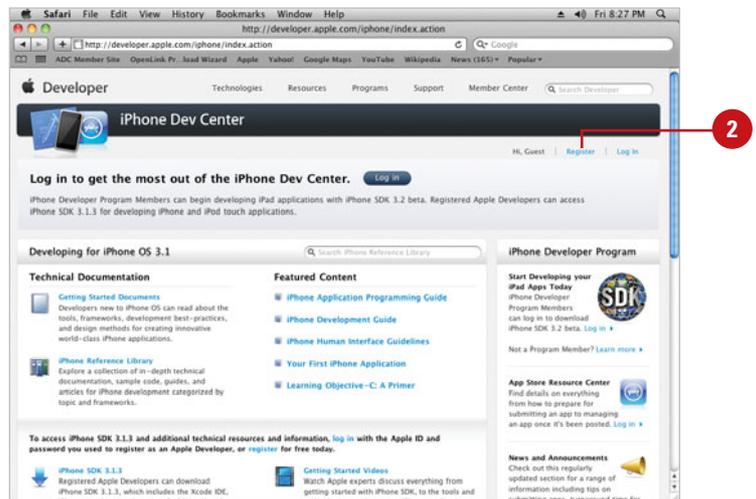
Use Multitouch in an App

Becoming an Apple Certified Developer

Become an Apple Certified Developer

- 1 Open your Web browser on the Mac, and then go to the iPhone Dev Center site:
 - ◆ <http://developer.apple.com/iphone>
- 2 Click the **Register** link on the Web page.
- 3 Follow the registration process; either use an existing iTunes Account or create a new account to register with the site.
 - ◆ Upon completion, you'll receive an email confirmation.
- 4 Go to <http://developer.apple.com/iphone/manage/certificates> to download and install your Apple iPhone Development Certificate.
- 5 Open the Keychain to view your Apple iPhone Development Certificates: a Developer and a Distribution Certificate.

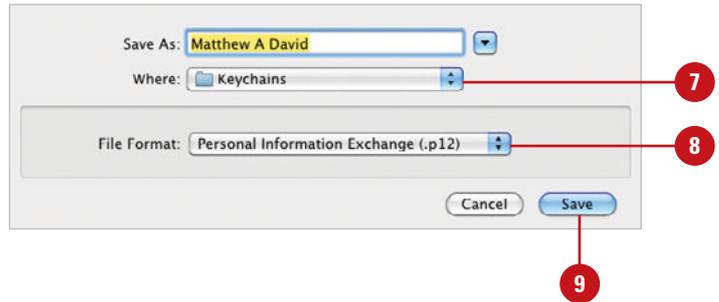
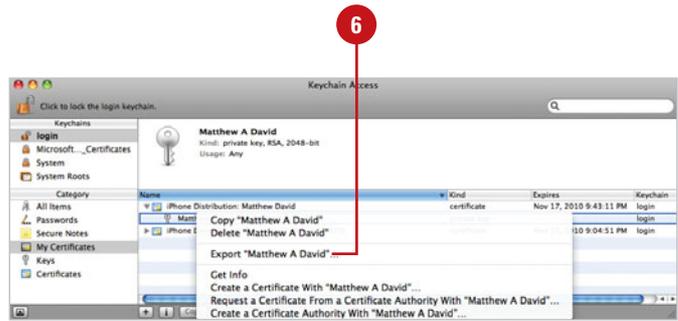
The iPhone OS has rapidly become the mobile operating systems against which all other mobile systems, such as Android, Windows Mobile 7, BlackBerry, WebOS and Symbian, are all compared to. There is a very good reason for this: the iPhone OS is rock solid. Each App you develop for the iPhone OS will run on an ever increasing number of products shipping from Apple. They include the iPhone 3G and 3GS, iPod Touch and the iPad. All of your Flash CS5 applications (**New!**) will run on these devices. To be able to develop applications for the iPhone you do need to become a registered Apple developer. The cost is only \$99 a year and this gives you the ability to load applications up to the iTunes App Store. Yes, you heard me right; it only costs \$99 a year to load as many apps as you can develop into the iTunes App Store. This is an amazing deal. Similar programs run by Nintendo to develop apps for their Wii Wares store start in the low thousands of dollars.



- 6 Control-click on the Developer Certificate, and then click **Export Certificate**.
- 7 Navigate to the location where you want to save the certificate.
- 8 Click the **File Format** list arrow, and then click **Personal Information Exchange (.p12)**.
- 9 Click **Save**.
- 10 Enter a strong password for your App, and then remember it as you'll need it later.
- 11 Repeat Steps 6-10 to export the Distribution Certificate in the Personal Information Exchange (.p12) format.

Did You Know?

You can use the P12 Certificates on Windows. You can copy the P12 Developer and Distribution Certificates to a disc and use them on a Windows computer. Creating the P12 Personal Information Exchange Certificates can be done on a Windows computer, but it requires the use of third party software such as OpenSSL. It is not easy. You are much better off starting with a Mac (find a friend, if you need to) and then moving to a Windows computer for your future Flash CS5 iPhone development needs.



For Your Information

Developing Apps on Macs and Windows

Amazingly, Adobe is the only company that allows you to develop iPhone Apps on Windows. All you need to accomplish this is to use a P12 Developer and Publishing certificate. The easiest way to get a P12 certificate is from a Mac computer. Just ask a friend if you can create an account on their Mac and then register yourself with Apple's Development program and create your own P12. After that you can copy the P12 certificate onto your Windows machine and develop the iPhone Apps.

Registering an iPhone and Other Apple Testing Devices

Find your iPhone's Unique Identifier

- 1 Connect your test iPhone/iPod Touch to your computer.
- 2 Open iTunes, and then select your device from the list to view the Summary tab information.
- 3 Click the Serial Number next to your device.

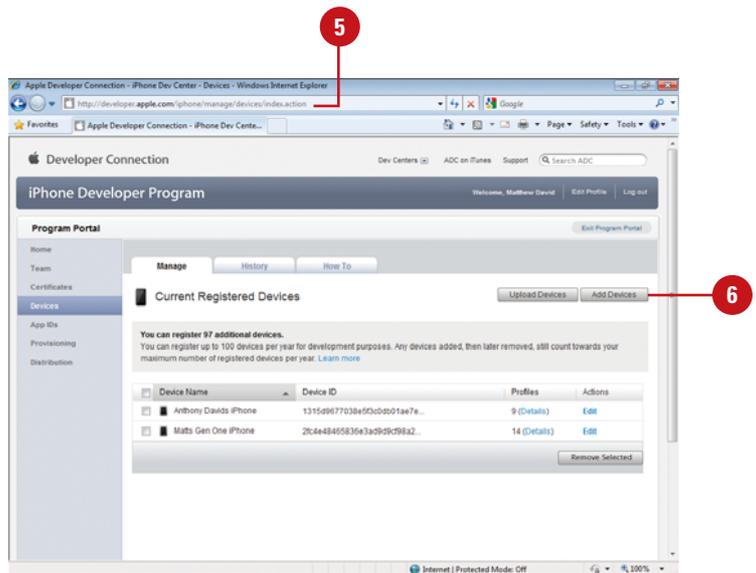
The number changes to an Identifier (UDID) number, which is about 40 alpha-numeric characters long.

- 4 Copy the UDID number.
- 5 Open your Web browser, and then go to the Devices page on the iPhone Developer site:

◆ <http://developer.apple.com/iphone/manage/devices/index.action>

- 6 Click the **Add Devices** button.
- 7 Paste in the UDID number and then enter a name for your device.
 - ◆ Suggestion: you may want to choose something like “iPhone Development” or “iPod Touch Gen 3 Development.”

The hardest part of developing apps for the iPhone is not the code you develop, but ensuring you have all of your i's and t's dotted and crossed per Apple's rigorous registration process. One small step in the wrong direction and your App will not compile or install correctly. It is worth taking your time to go through the setup process (**New!**). After you have your P12 Certificates you will need to register your test iPhone with Apple. You can register up to 100 devices per year. The developer account ID, registered device and P12 certificates are needed for all apps you want to sell in the iTunes App Store. You can register iPhone's, iPod Touches and iPads.

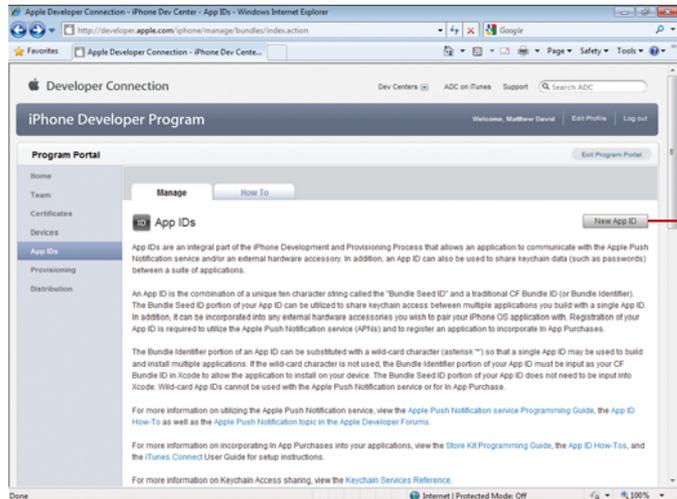


Creating App ID's

Create an App ID

- 1 Open your Web browser, and then go to the App ID page on the iPhone Developer site:
 - ◆ <http://developer.apple.com/iphone/manage/bundles/index.action>
- 2 Click the **New App ID** button.
- 3 Enter a new App name you can find easily, such as *WorldlyWordSearch*; the name has to be all one word.
- 4 Enter a Bundle Seed ID.
 - ◆ The convention is to reverse your Web site address and add your Apps name at the end. For example, *com.pearson.worldlywordsearch*; the name has to be all one word.
- 5 Click **OK**.

The App ID is a unique identifier Apple uses to identify your App from the tens of thousands of Apps in the iTunes App Store. There are ten's of thousands of applications in the iTunes App Store with more in constant development. Part of your registration process is to assign your App an ID, called the App ID. The App ID (**New!**) will be connected to your certificate and to the development and publication profiles you generate. You control the name of the App, but behind the scenes, Apple associates it an unique ID, called a Bundle Seed ID. Apple will know what your App is when it comes time for you to submit your App to iTunes.



Creating a Developers Provisioning Profiles

Create a Developer Profile

1 Open your Web browser, and then go to the Provisioning Profiles page on the iPhone Developer site:

◆ <http://developer.apple.com/iphone/manage/provisioningprofiles/index.action>

2 Click the **Development** tab.

3 Click the **New Profile** button.

4 Enter a meaningful Profile name.

A convention that is gaining popularity is to use the prefix "Dev" followed by the App ID, such as *DevWorldlyWordSearch*.

5 Select the check box with your name in the Certificates list.

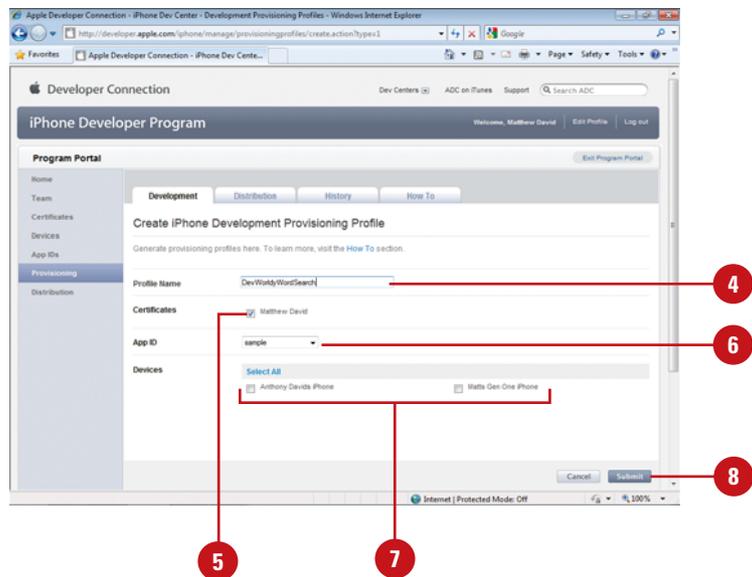
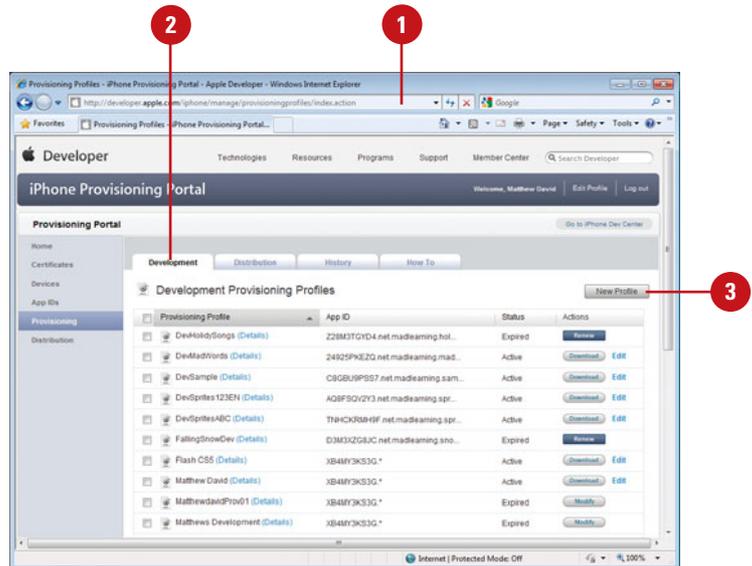
6 Click the **App ID** list arrow, and then select your App ID.

7 Select your test device from the devices listed.

8 Click the **Submit** button.

Your developer profile takes about 30 seconds to generate.

A provision profile is a document which is associated with your App and developer certificate for either development or iTunes App distribution. There are three types of Provisioning Profiles (**New!**) you can use: Developer, Distribution to Ad Hoc and Distribution to iTunes App Store. You will cover how to use the Distribution Profiles later in the chapter. For now, let's focus on creating Developer Profiles.



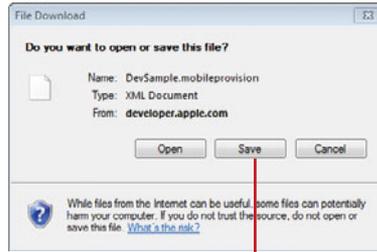
- 9 Click **Download** (Mac) or **Save** (Win), and then save the Developer profile to your desktop.

The file will have the extension “mobileprovision”

- 10 Connect your test iPhone to iTunes.

- 11 Drag the downloaded Developer Profile onto iTunes, and then sync your iPhone.

This adds the Developer Profile to your testing device.



For Your Information

Developing Icons for your iPhone Apps

In preparation for your iPhone App there are four PNG images you need:

29.png – a file that is 29x29 pixels

57.png – a file that is 57x57 pixels

512.png – a file that is 512x512 pixels

Default.png – this placeholder file that is used while your App is loading. It is important the file starts with a capital “D”.

You can create all of these files using Adobe Fireworks.

Creating and Publishing an iPhone App

Create and Publish an iPhone App

- 1 In Flash, click the **File** menu, click **New**, click **iPhone OS**, and then click **OK**.

A new movie opens with the size 320x480 pixels.

- 2 Create an iPhone App using Flash tools.

- 3 Open the **Properties** panel.

- 4 Click the iPhone Settings **Edit** button.

The iPhone Settings dialog box opens, displaying three tabs: General, Deployment and Icons.

- 5 Click the **General** Tab.

- 6 Enter an Output file name, such as *wordsearch.ipa*.

- 7 Enter the App Name that you want to appear on the iPhone, such as *Word Search*. You are limited to 11 characters.

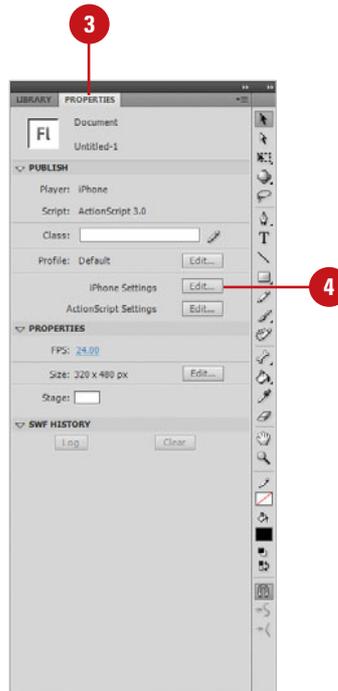
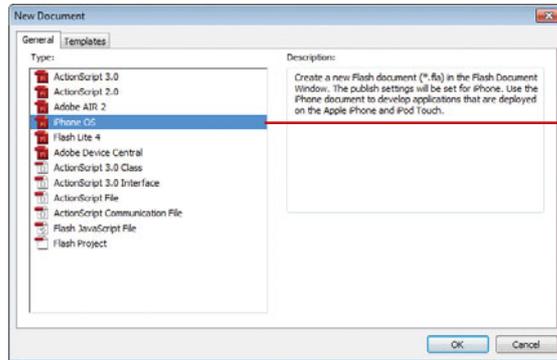
- 8 Enter a version number for the App, such as *1.0*.

- 9 Click the **Aspect Ratio** list arrow, and then select a display option: Portrait or Landscape view.

- 10 Select or deselect any of the following options:

- ◆ **Full Screen**. Select to force your App to take up all of the screen space on your phone;

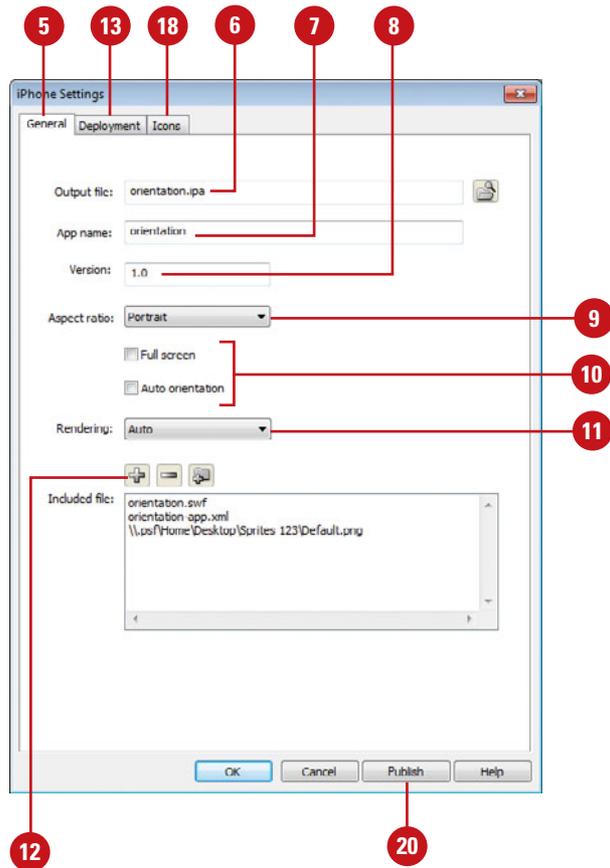
With all of these tasks completed you are now ready to complete your first iPhone App. Sometimes it seems like a world full of red tape where you need to dot your i's and cross your t's, but you do get to the point where you can develop applications for your iPhone using Flash CS5 (**New!**). You need to go through the steps of creating a Developer Certificate, registering your test iPhone, creating an App ID, and downloading a developers profile in order to develop for any iPhone development tool; this is not just a unique Flash CS5 feature. Flash CS5, however, does allow you to very easily create the final iPhone App.



deselect to show information at the top of your iPhone, including carrier signal, WiFi connection, time and battery life, will be visible in your App.

- ◆ **Auto Orientation.** Select to change the orientation of your App as the iPhone is rotated.

- 11 Click the **Rendering** list arrow, and then click an acceleration option, **Auto**, **CPU**, or **GPU**.
 - 12 Click the **Add** button (+) in Included Files, and then select the Default.png file. The Default.png file displays when your App starts to run and disappears when your App has loaded.
 - 13 Click the **Deployment** tab.
 - 14 Click **Browse**, and then select your Developer P12 certificate file, and then enter your password.
 - 15 Click **Browse**, and then select the developer Provisioning Profile you downloaded.
 - 16 Enter the full name of the App ID you created in the iPhone Developer Center.
 - 17 Select the **Quick Publishing For Device Testing** option.
 - 18 Click the **Icons** tab.
 - 19 Locate and add the 29.png, 57.png and 512.png file.
 - 20 Click **Publish**.
- Flash launches the iPhone Packager tool and generates (6-10 minutes) an IPA iPhone App file in the same folder as your Flash files.
- 21 Locate the IPA file, and then drag the file onto iTunes and sync your device. When the sync completes you will be able to launch your App from your iPhone.



For Your Information

Playing Back Video

You can play back video in the iPhone, but there is a caveat. The only types of video that you can add to your iPhone Apps are Sorenson and ON2 VP6 video formats. If you want to use H.264 then you need to essentially post the video to a URL string so it pops up in Mobile Safari.

Controlling the Accelerometer

With the release of the Flash Player 10.1 and Adobe Integrated Runtime, AIR 2.0, the Flash team added several new core API features (**New!**). Access to a device's Accelerometer is one of those. The role of the Accelerometer is to detect when you move your phone. The Accelerometer is a listener that is triggered when it is used. The following example adds an Accelerometer listener to your iPhone App.

Add an Accelerometer Listener

- 1 Create a new iPhone App and add the necessary development properties in the iPhone settings.
- 2 Add a dynamic text field to the Stage with the name *myTextField* in the Properties panel.
- 3 Create a new layer on the Timeline with the named *Actions*, and then select the Actions layer.
- 4 Open the **Actions** panel.
- 5 Add code to import the libraries for the Accelerometer to work correctly:

```
import flash.events.AccelerometerEvent
import flash.sensors.Accelerometer;
```

- 6 Add code to create a new Accelerometer object:

```
var acc1:Accelerometer = new Accelerometer()
```

- 7 Add a boolean object to test if the Accelerometer works or not:

```
var isSupported:Boolean =
    Accelerometer.isSupported;

    checksupport();
```

- 8 Add a function that contains the event listener, which waits for the Accelerometer to be triggered:

```
function checksupport():void {
    if (isSupported) {
        myTextField.text = "Accelerometer feature
            supported";
        acc1.addEventListener
            (AccelerometerEvent.UPDATE, updateHandler);
    } else {
        myTextField.text = "howdy ";    }
}
```

- 9 Add a function that posts a message to the text field to tell what direction the device has moved to:

```
function
    updateHandler(evt:AccelerometerEvent):void {
myTextField.text = String("at: " + evt.timestamp
    + "\n" + "acceleration X: " + evt.accelerationX
    + "\n" + "acceleration Y: " + evt.accelerationY
    + "\n" + "acceleration Z: " +
    evt.accelerationZ);
```

- 10 Publish and package your file into an iPhone App and test it on your iPhone.

The Accelerometer gives you new ways for your customers to interface with your applications beyond touchscreen controls. The Accelerometer works great on the iPhone but the same code can be used for Adobe AIR apps running on Google's Android OS, Palm's WebOS and RIM's BlackBerry phones. Yes, that's right. Develop one App and have it deployed to multiple mobile devices.

Saving Images to the Camera Roll

Adobe does give you access to some core iPhone specific tools. One of those is the ability to add a function that will save an image of the screen to the Camera Roll (**New!**). The following example saves a screen image to the Camera Roll in an iPhone App.

Save a Screen Image to Camera Roll

- 1 Click the **File** menu, click **New**, click **iPhone OS**, and then click **OK**.
- 2 Create a new Movie Clip on the Stage with the name *snapShot*.
- 3 Add the following event Listener to:

```
snapShot.addEventListener(MouseEvent.CLICK, myScreenShot);
```

- 4 Add the following function that takes a screen shot of your iPhone:

```
function myScreenShot  
(event:MouseEvent):void
```

```
{
```

```
if (CameraRoll.supportsAddBitmapData)
```

```
{
```

```
var cameraRoll:CameraRoll = new  
CameraRoll();  
cameraRoll.addEventListener(ErrorEvent.ERROR  
, onCrError);
```

```
cameraRoll.addEventListener(Event.COMPLETE,  
onCrComplete);
```

```
var bitmapData:BitmapData = new  
BitmapData(stage.stageWidth,stage.stageHeigh  
t);
```

```
bitmapData.draw(stage);
```

```
cameraRoll.addBitmapData(bitmapData);
```

```
}
```

```
else
```

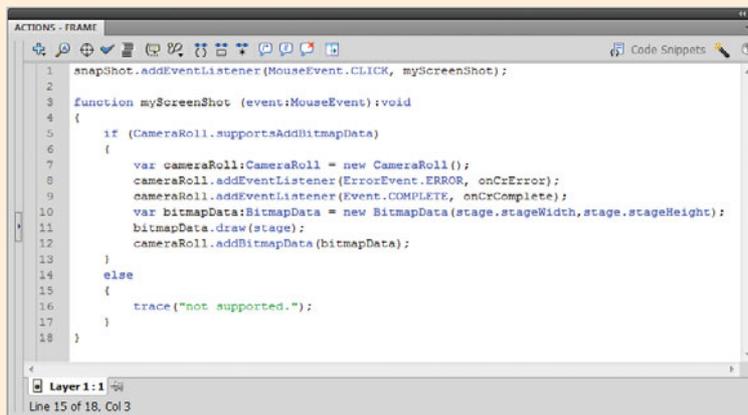
```
{
```

```
trace("not supported.");
```

```
}
```

```
}
```

- 5 Publish and package your file into an iPhone App and test it on your iPhone.



```
ACTIONS - FRAME  
1 snapShot.addEventListener(MouseEvent.CLICK, myScreenShot);  
2  
3 function myScreenShot (event:MouseEvent):void  
4 {  
5     if (CameraRoll.supportsAddBitmapData)  
6     {  
7         var cameraRoll:CameraRoll = new CameraRoll();  
8         cameraRoll.addEventListener(ErrorEvent.ERROR, onCrError);  
9         cameraRoll.addEventListener(Event.COMPLETE, onCrComplete);  
10        var bitmapData:BitmapData = new BitmapData(stage.stageWidth,stage.stageHeight);  
11        bitmapData.draw(stage);  
12        cameraRoll.addBitmapData(bitmapData);  
13    }  
14    else  
15    {  
16        trace("not supported.");  
17    }  
18 }
```

Layer 1:1
Line 15 of 18, Col 3

Understanding the Limits of Flash

There are some limitations to using the iPhone Packager for Flash CS5 (**New!**). The first is that you cannot load external SWF movies. You can, however, load external JPG and PNG files and sound files such as MP3.

There are also performance problems between different versions of the iPhone. For instance, the iPhone 3GS is literally twice as fast as the iPhone 3G and original iPhone. There are some simple tricks you can do to speed up things. The following code will prevent event Bubbles starting:

```
override public function
    dispatchEvent(evt:Event):Boolean {
    if (hasEventListener(evt.type) || evt.bubbles) {
    return super.dispatchEvent(evt);
    }
    return true;
    }
```

A second optimization trick you can do is to restrict your use of vector based images inside of Flash. Use PNG formatted images where possible. The good news is that the iPhone has great support for PNG files.

If you do need to use vector images you can fool the iPhone into thinking it's a bitmap image by using the `cacheAsBitmapMatrix`.

- 1 Create a new image. Add code import the Flash Geom Matrix:

```
import flash.geom.Matrix;
```

- 2 Create a new shape:

```
var my_shape :MyShape = new MyShape();
addChild(my_shape);
```

- 3 Add the `cacheAsBitmap` property to ensure that all objects that create are cached:

```
my_shape.cacheAsBitmap = true;
my_shape.cacheAsBitmapMatrix = new
Matrix();
```

- 4 Create images on the screen that the iPhone thinks are bitmaps.

A final significant limitation is the use of audio files. Audio files can have a short delay between an event happening and the sound playing. This is because the audio file is not in the iPhone's cache for playback. You can avoid this by exporting your audio file to be triggered in the first frame of your movie.

What Flash Does Not Allow You To Do

Adobe has gone to great lengths to allow you to build applications for the iPhone using traditional Flash tools. BUT (and it is a big BUT) there are some things that Flash does not allow you to do using the iPhone Packager.

The biggest challenge you will have is the inability to use ActionScript 1.0 and 2.0. All of your apps must be developed using ActionScript 3.0. There are tens of thousands of great ActionScript 1.0 and 2.0 apps that can not be published to the iPhone due to the ActionScript 3.0 limitation.

You also have to be careful using video. Only Flash Video will work. You can use a URL link to load a MPEG4 video, but this is really a cheat as you are running the video through the iPhone's Mobile Safari Web browser.

If you are familiar developing iPhone Apps using the iPhone SDK then you may expect core iPhone Cocoa Touch kits available to you in Flash. This includes the StoreKit (for "in App purchase"), GameKit, MapKit and other rich media tools. These useful kits are simply not available to the Flash developer.

The good news is that Adobe is well aware that the Flash CS5 iPhone Packager is not complete. Expect releases for the iPhone Packager to keep coming during the lifecycle of Flash CS5.

Identifying Devices to Deploy an Ad Hoc App

Collect and Add Deployment Device IDs

1 Contact each person you will be deploying your App to and ask them to connect their iPhone, iPod Touch or iPad to their Mac or Windows computer.

2 Ask them to open iTunes, select their device, and then view the Summary tab information.

3 Ask them to click the Serial Number next to their device.

The number changes to an Identifier (UDID) number.

4 Ask them to copy the UDID number, and then send it to you.

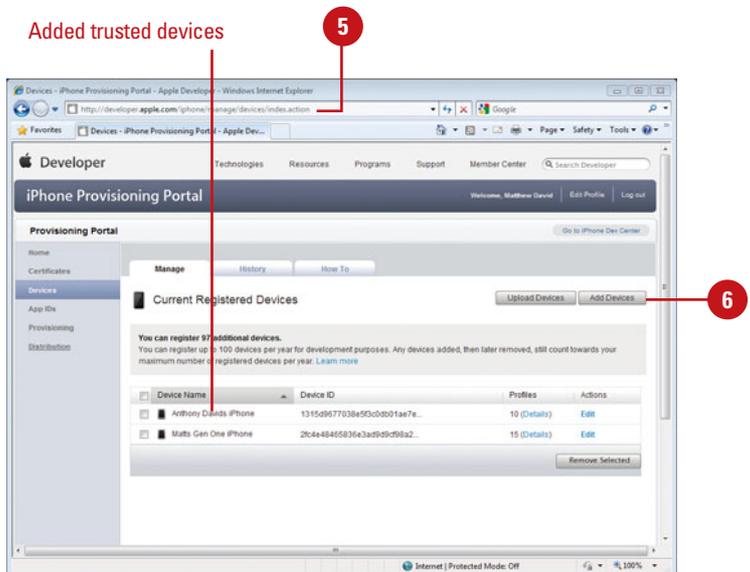
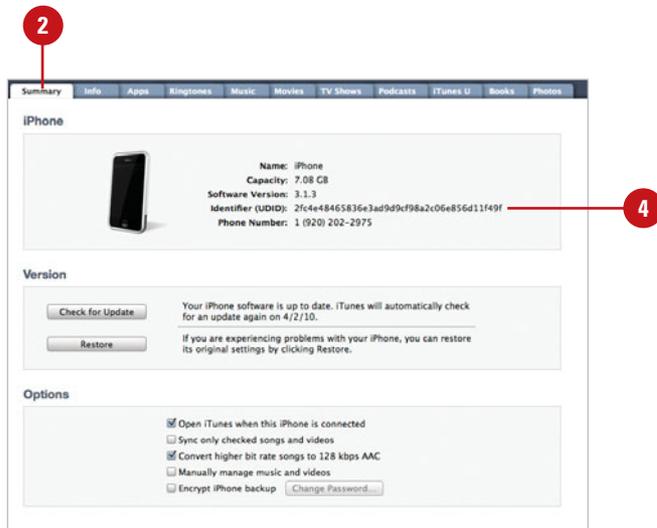
5 Open your Web browser, and then go to the Devices page on the iPhone Developer site:

◆ <http://developer.apple.com/iphone/manage/devices/index.action>

6 For each trusted device, click the **Add Devices** button on the Web page, paste in the UDID number and then enter a device name.

IMPORTANT You can only have 100 identified devices per year. You cannot add and remove devices to keep your list at 100. When you have added the 100th device you have to wait a year before you can remove a device and add a new one.

Ad Hoc is a method for deploying your App to up to 100 iPhone, iPod Touch and iPad devices. The Ad Hoc method (**New!**) is a solution that allows you to deploy your apps to a small group of friends and co-workers without the App appearing in the iTunes App Store. The method for creating the application is very similar to deploying an App for deployment to the iTunes App store. There are three steps you need to be able to successfully deploy an App using Ad Hoc: Identify each device you will be deploying your Ad Hoc App too; creating an Ad Hoc provisioning license; and packaging your Ad Hoc App.



Creating an Ad Hoc Distribution Profile

Create an Ad Hoc Distribution Profile

1 Open your Web browser, and then go to the Provisioning Profiles page on the iPhone Developer site:

◆ <http://developer.apple.com/iphone/manage/provisioningprofiles/index.action>

2 Click the **Distribution** tab.

3 Click the **New Profile** button.

4 Click the **Ad Hoc** option.

5 Enter a profile name for your App.

6 Click the **App ID** list arrow, and then select the App ID that matches the App you want to deploy using Ad Hoc.

7 Select the devices (from the list of ones you entered) you want to deploy using Ad Hoc.

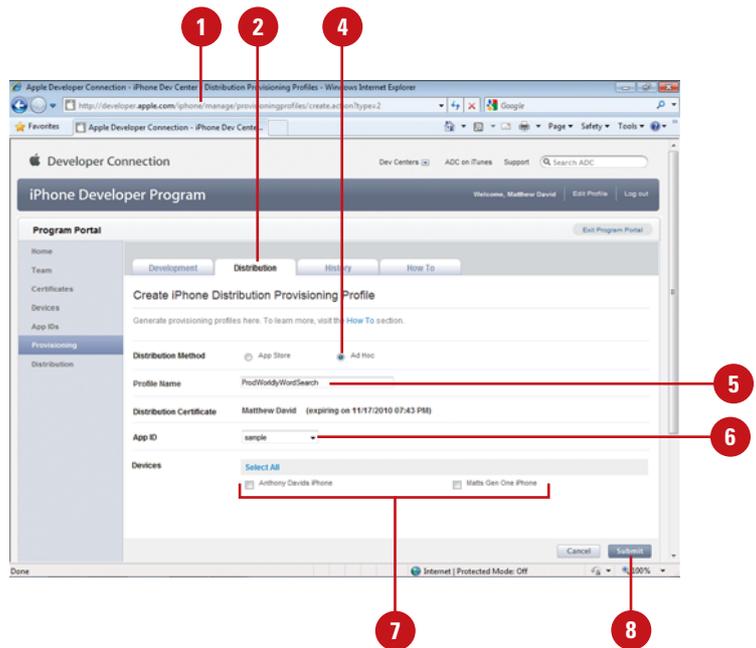
8 Click the **Submit** button.

Your Ad Hoc profile takes about 30 seconds to generate.

9 Click **Download** (Mac) or **Save** (Win), and then save the Ad Hoc profile to your desktop.

At this point, you're ready to create your Ad Hoc iPhone App.

The first step is to identify the devices you will be deploying your Ad Hoc App to. The next step is to create the profile you will need to use when you are building your Application for Ad Hoc deployment. The Ad Hoc Distribution method (**New!**) removes the hassle of having to wait for your App to be approved by Apple (which can take days to weeks to complete). To be able to take advantage of the Ad Hoc deployment process you need to first create an Ad Hoc Distribution Profile. Creating the Ad Hoc Profile is very similar to the profile you will create when you submit your App to the iTunes Store. The one big difference is you need to have a list of all the unique iPhone's and iPod Touch devices you will be sending your final App too.



Packaging an Ad Hoc App

Publish and Package an Ad Hoc App

- 1 In Flash, open the iPhone App you want to publish and package as an Ad Hoc App.
- 2 Open the **Properties** panel.
- 3 Click the iPhone Settings **Edit** button.

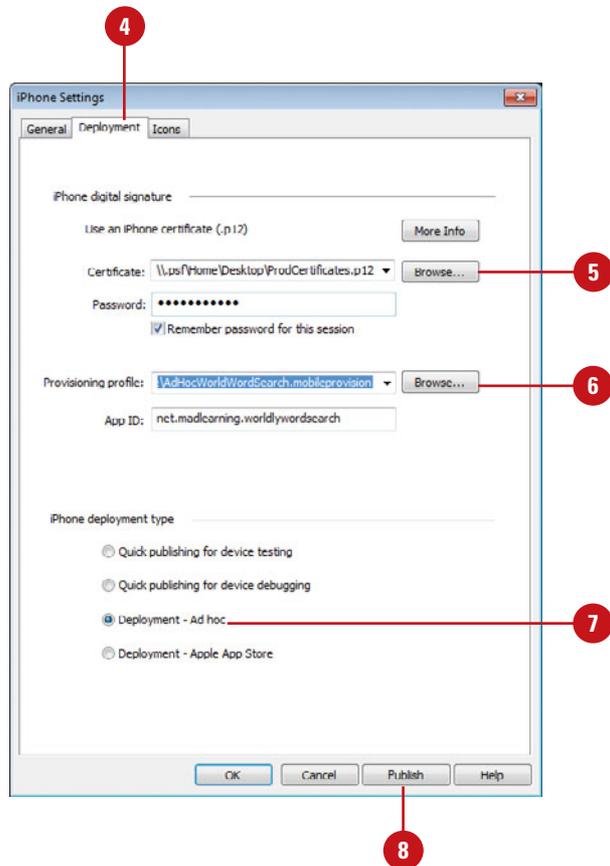
The iPhone Settings dialog box opens, displaying three tabs: General, Deployment and Icons.

- 4 Click the **Deployment** tab.
- 5 Click **Browse**, and then select your Published P12 certificate file, and then enter your password.
- 6 Click **Browse**, and then select the Ad Hoc Distribution Profile you downloaded.
- 7 Click the **Deployment – Ad Hoc** option.
- 8 Click **Publish**.

Flash launches the iPhone Packager tool and generates (6-10 minutes) an IPA iPhone App file in the same folder as your Flash files.

- 10 Locate the IPA file and send it to anyone in the Ad Hoc Provisioning Profile list. Ask them to drag the file onto iTunes and sync their device. When the sync completes you will be able to launch your App from your iPhone.

The final step you need to take is to package your Flash movie into an iPhone App using the Ad Hoc profile (**New!**). The final step in creating your Ad Hoc App is to package your Flash App for distribution. The Ad Hoc distribution process is very similar to the final App Store Packaging publication process. There are some differences. The main change is that you must use the Ad Hoc Profile you created in the previous section. Without the Ad Hoc Profile your Ad Hoc deployment method will not work correctly.



Packaging an App for the iTunes App Store

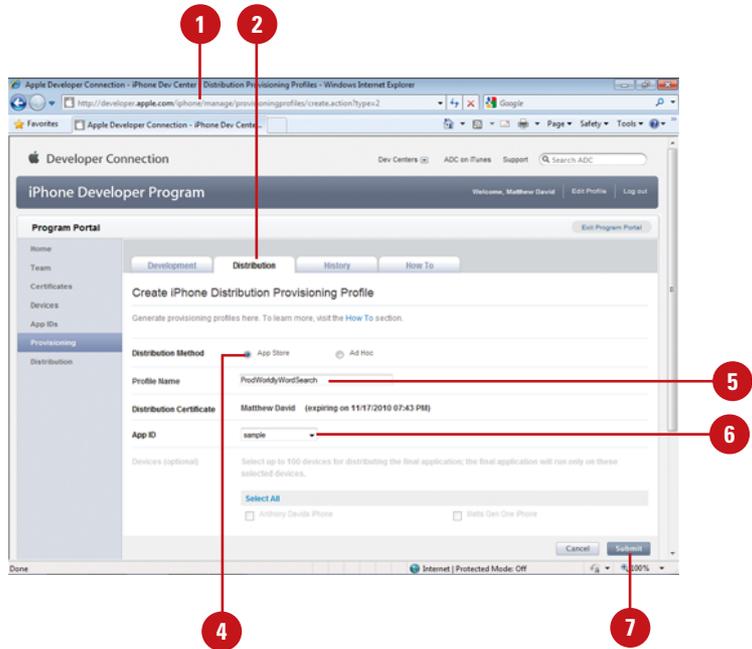
Publish and Package an iPhone App

- 1 Open your Web browser, and then go to the Provisioning Profiles page on the iPhone Developer site:
 - ◆ <http://developer.apple.com/iphone/manage/provisioningprofiles/index.action>
- 2 Click the **Distribution** tab.
- 3 Click the **New Profile** button.
- 4 Click the **App Store** option.
- 5 Enter a profile name for your App.
- 6 Click the **App ID** list arrow, and then select the App ID that matches the App you want to deploy.
- 7 Click the **Submit** button.

Your profile takes about 30 seconds to generate.
- 8 Click **Download** (Mac) or **Save** (Win), and then save the App Store profile to your desktop.
- 9 In Flash, open the iPhone App you want to publish and package as an App Store App.
- 10 Open the **Properties** Panel.
- 11 Click the iPhone Settings **Edit** button.

The iPhone Settings dialog box opens, displaying three tabs: General, Deployment and Icons.

Apple's iTunes Store is an amazing success story. 133,000 Apps and 3 Billion downloads in 20 months is nothing to sneeze at. There are many stories of companies making millions from Apple. But, before you can get any money, you need to have your App ready for deployment. The method for packaging your Flash file into an iPhone App for distributing to the iTunes store is very similar to the Ad Hoc process with the exception that you only need to create a distributing profile and create your iPhone IPA file from Flash CS5 (**New!**).



- 12 Select the **Deployment** tab.
- 13 Click **Browse**, and then select your Published P12 certificate file, and then enter your password.
- 14 Click **Browse**, and then select the App Store Distribution Profile you downloaded.
- 15 Click the **Deployment – Apple App Store** option.
- 16 Click **Publish**.

Flash launches the iPhone Packager tool and generates (6-10 minutes) an IPA iPhone App file in the same folder as your Flash files.

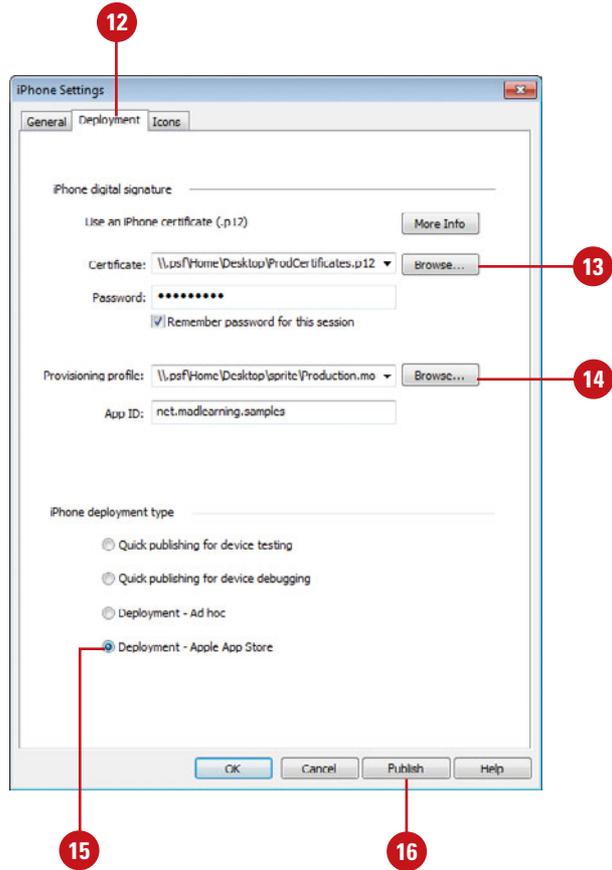
- 17 Locate the IPA file and then change the extension of the file from IPA to ZIP.

Both file formats are container formats. That is, they contain all of the files needed for the App to run; iTunes Connect only accepts files in the ZIP format.

At this point, your App is ready for deployment to the iTunes App Store.

Did You Know?

Ad Hoc is a great way to easily deploy apps to a small group of users. It is not a good solution for deploying apps to a large number of users. If you want to deploy your App to a large number of users then you will need to take advantage of Apple's powerful iTunes App Store.



Using iTunes Connect to Publish an App

Use iTunes Connect to Publish your App

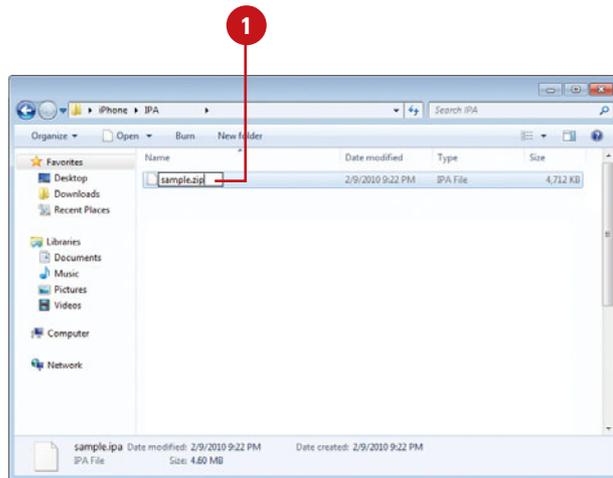
- 1 In Flash, create your iPhone App, publish and package it, and then change the extension from IPA to ZIP.
- 2 Convert the 512x512 PNG pixel image into a JPG image. Label the new file 512.jpg.
- 3 Take a screen shot of your App as it appears in your iPhone. When your App is playing on your iPhone, press the Home and Sleep buttons at the same time (screen flashes) to take a screen shot of your App.

The image is stored to your Camera Roll and is exactly the same size Apple needs.

Did You Know?

You can download Apple's *Developer Guide*. Open your Web browser, go to the Manage Your Application page on iTunes Connect, and then click the download link to save the PDF file.

You are now very close to having an App available on the iTunes store. Can't you feel the rush! You could be selling thousands of apps in a matter of a few days. The gap between you and riches is Apple's iTunes Connect publishing tool. You are very close now. You will be using a new Web site to upload your final iPhone Apps. The site is called iTunes Connect (<https://itunesconnect.apple.com/>). In every sense, iTunes Connect is your business relationship with Apple. The site allows you to set up your contracts, tax records, banking information, review sales trends, download financial report and manage your In App Purchases. You will need to complete these sections in order to sell your App in iTunes. This section is going to focus on the important part of iTunes Connect: Managing your Applications (**New!**).



4 Open your Web browser, and then go to the iTunes Connect page, and then use your Apple Developer ID and Password to log into the site:

◆ <https://itunesconnect.apple.com>

5 Click **Manage Your App** from iTunes Connect.

6 Add a new application or edit an existing one.

◆ **Add New App.** Click the **Add New Application** button to start the process of creating a new iTunes App.

◆ **Edit Existing App.** Double-click the application icon.

When you're done editing your App settings, click **Save Changes**.

There are several screens you need to complete to upload your App to iTunes.

7 If prompted, click the **Yes** or **No** option to specify whether your App contains encrypted data.

◆ For a first App, click **No** to keep the process simple. In the future, you can experiment with different settings.

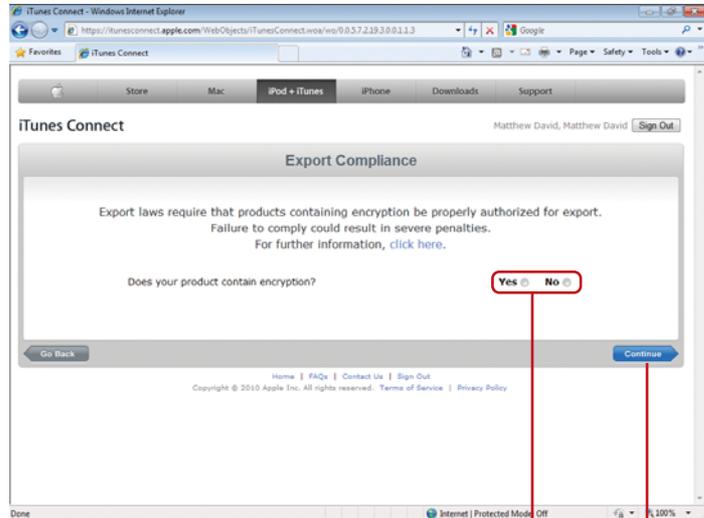
8 Click the **Continue** button when you have completed the page.

The next page allows you to add product overview information about your App. For the most part, the content you enter on this screen can be edited after your App has been submitted.

Continue Next Page



Link to download the Developer Guide



Continued from Previous Page

9 Enter or change information about your App. The information includes the following:

- ◆ Name and description.
- ◆ Product primary and secondary categories.
- ◆ Copyright, version number, and SKU number.
- ◆ Keywords. Use for search engines (up to 100 characters).
- ◆ URLs. Application/support URL and support email.

10 Click the **Continue** button when you have completed the page.

The next page allows you to add a rating for your App.

11 Click options to answer the 10 questions to determine a rating.

12 Click the **Continue** button when you have completed the page.

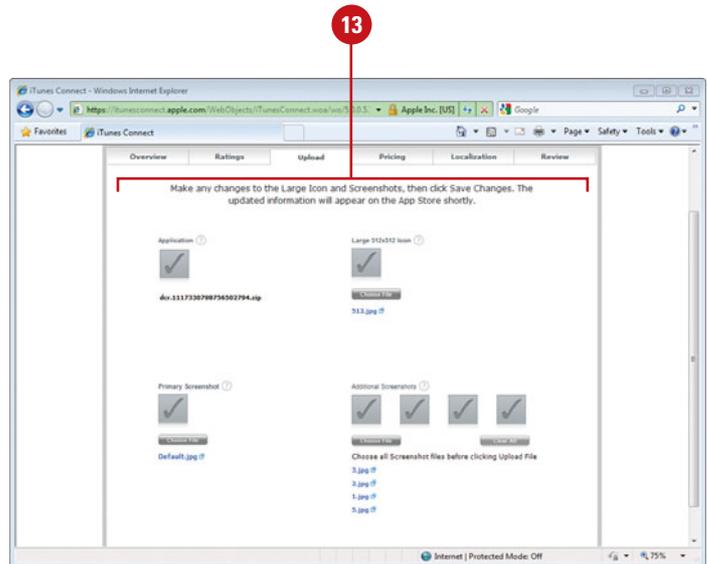
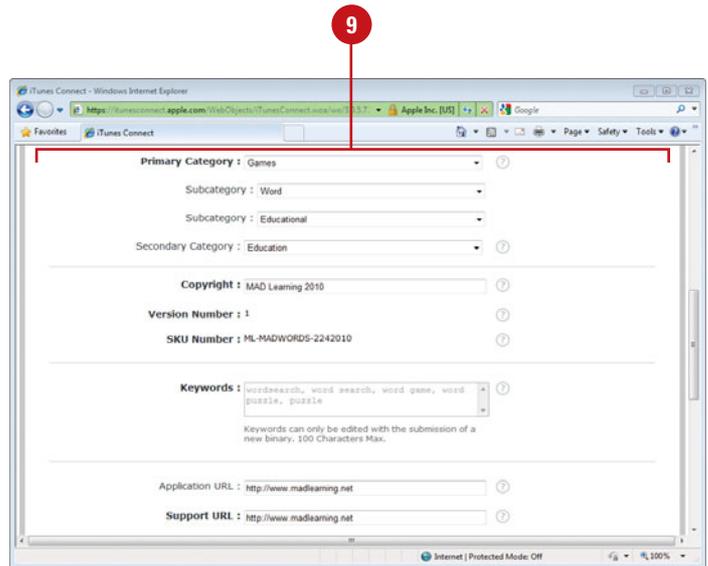
The next page allows you to upload all of your App files.

13 Click the **Choose File** button to upload your App files:

- ◆ Upload the ZIP'd IPA file for the Application.
- ◆ Upload the 512.jpg for the Large 512x512 Icon.
- ◆ Upload a 480x320 jpg image for the Primary Screenshot.
- ◆ Add 1-4 480x320 jpg images for the Additional Screenshots.

The upload will take about 20 minutes. The upload is checking for some basic settings such as including the correct profile.

A green check mark appears for each successfully loaded image. A red cross appears when something is wrong.



- 14 Click the **Continue** button when you have completed the page.

The next page allows you to specify availability and pricing for your App.

- 15 Specify an Availability Date and Price Tier.

- ◆ **Availability Date.** Specifies when your App goes on sale.
- ◆ **Price Tier.** Select a price tier level; you do not get to select a specific price for your App.

This removes the pain of selling with different currencies. A Tier 2 App will be \$0.99 in the US, 59 pence in England and AU\$1.29 in Australia. There are over 70 different currencies that Apple manages for you.

- 16 Click the **Continue** button when you have completed the page.

The next page allows you to specify localization for your App.

- 17 Specify the localization settings you want for your App.

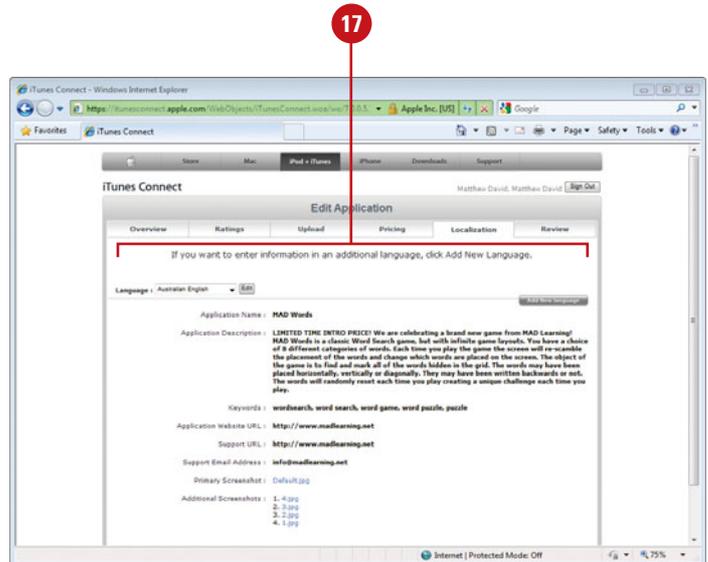
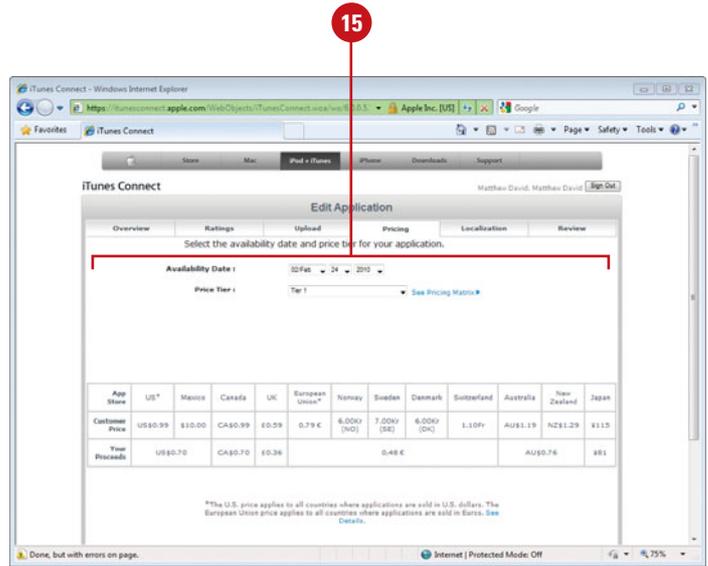
- 18 Click the **Continue** button when you have completed the page.

- 19 At this point, review all of your content, and then click the **Submit** button.

- ◆ When you're done editing your App settings, click **Save Changes**.

Did You Know?

Faster iTunes App Submission. Apple has done a lot to improve the review process for new apps. Today it only takes 5-7 days for an App to be approved and appear in iTunes.



Using Screen Orientation in an App

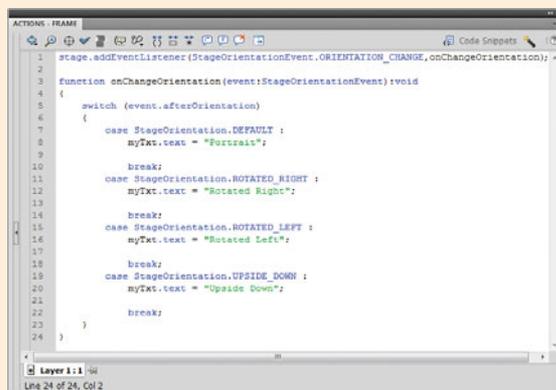
The iPhone has one characteristic that you will not see on a desktop or laptop computer: it can be easily rotated. Determining orientation can allow you to add interesting effects to your iPhone Apps (**New!**). For instance, the default Stocks tool on your iPhone will show you the numbers of a particular stock when you hold your phone in portrait; switch the phone to landscape and the content changes to a line chart illustrating how the stock is performing. This trick is accomplished through the iPhone being able to detect the orientation of the device. The iPhone can detect Orientation in four distinct directions:

- ◆ Default – Portrait
- ◆ Rotation Left – Landscape
- ◆ Rotation Right – Landscape
- ◆ Upside Down – Portrait, upside down

ActionScript contains a special class called the Orientation Class, which allows you to control content on the screen depending on the orientation of the device.

Use the Screen Orientation Class

- 1 Create a new iPhone App, apply the appropriate provisioning profiles, and then save the file with the name *iPhoneOrientation.fla*.
- 2 Add a dynamic text box to the Stage with the name *myTxt*, and then apply the *_Sans* font.
- 3 Open the **Actions** Panel.
- 4 Add a listener and function that changes the content of the text box based on the orientation of the device (see illustration).
- 5 Publish and package your file into an iPhone App and test it on your iPhone. Rotate your iPhone to see the text change on your iPhone.



```
1 stage.addEventListener(StageOrientationEvent.ORIENTATION_CHANGE, onChangeOrientation);
2
3 function onChangeOrientation(event:StageOrientationEvent):void
4 {
5     switch (event.afterOrientation)
6     {
7         case StageOrientation.DEFAULT :
8             myTxt.text = "Portrait";
9
10            break;
11            case StageOrientation.ROTATED_RIGHT :
12                myTxt.text = "Rotated Right";
13
14                break;
15                case StageOrientation.ROTATED_LEFT :
16                    myTxt.text = "Rotated Left";
17
18                    break;
19                    case StageOrientation.UPSIDE_DOWN :
20                        myTxt.text = "Upside Down";
21
22                        break;
23            }
24 }
```

In addition to using ActionScript to control the orientation of your phone, you can also have the whole movie auto-orientate as a default setting in your iPhone Settings screen. On the whole, however, you have much tighter control over your content is you use ActionScript to control the orientation of specific movies, sprites and objects over the auto-orientation Boolean setting.

For Your Information

Loading Files Into the iPhone

The Loader Class allows you to load external content into your Flash Player. For the most part, this still works when you package your Flash movie for the iPhone. For instance, you can still load MP3, Sorenson Flash video and PNG files. There is, however, one exception: you cannot load SWF files that contain ActionScript into your movies. This is a pain as a lot of Flash designers like to break up their movies into smaller, individual SWF files and then you use a shell SWF to package it all together. Hopefully that will change in future releases.

Using Geolocation in an App

Mobile devices are meant to move. With GeoLocation services, you can figure out your current location. So, when someone asks you “Where are you?” you can tell them.

The following example shows how to add Adobe AIR 2.0 GeoLocation Class (**New!**) to your iPhone App. This sample is stripped down to a simple Class file, which can give you an idea of how you can use this in other Apps.

Add GeoLocation Services

1 Create a new iPhone App, apply the appropriate certificates and profiles, and then save your file with the name *iPhoneGeolocation fla*.

2 Add a dynamic ClassicText box to the Stage with the name *myTxt*, and then apply the *_Sans* font.

3 Create a new Class for the App with the name *iPhoneGeolocation*.

4 Save the ActionScript file with the name *iPhoneGeolocation.as*.

5 Add code to import the class files that allow you to use the Geolocation objects:

```
import flash.events.GeolocationEvent;
import flash.sensors.Geolocation
```

6 Create a new geolocation variable that you can use:

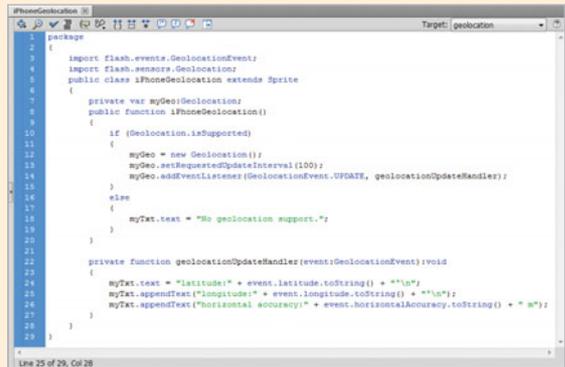
```
private var myGeo:Geolocation;
```

7 Create a simple If/Else statement that detects whether or not the geolocation services can be used.

The iPhone always gives the user of the iPhone a choice to whether or not you can use the Geolocation services. If you do not allow geolocation to be used a message will be posted to the text field

informing the user that a location could not be detected. If you do, however, allow the user to use geolocation then a new listener event will be triggered and then re-run 10x/second.

8 Add a function triggered by the geolocation listener:



```
package
{
    import flash.events.GeolocationEvent;
    import flash.sensors.Geolocation;
    public class iPhoneGeolocation extends Sprite
    {
        private var myGeo:Geolocation;
        public function iPhoneGeolocation()
        {
            if (Geolocation.isSupported)
            {
                myGeo = new Geolocation();
                myGeo.setRequestedUpdateInterval(100);
                myGeo.addEventListener(GeolocationEvent.UPDATE, geolocationUpdateHandler);
            }
            else
            {
                myTxt.text = "No geolocation support.";
            }
        }
        private function geolocationUpdateHandler(event:GeolocationEvent):void
        {
            myTxt.text = "Latitude" + event.latitude.toString() + "°N";
            myTxt.appendText("Longitude" + event.longitude.toString() + "°W");
            myTxt.appendText("Horizontal accuracy" + event.horizontalAccuracy.toString() + " m");
        }
    }
}
```

9 Publish and package your file into an iPhone App and test it on your iPhone.

How Does the iPhone Know Your Location?

There are three ways to detect your location. The most accurate method uses the Global Position Satellites (GPS) built into the iPhone 3G and 3GS. The second method uses Cell Tower Triangulation where the location of the device is determined by using 3 or more Cell Tower's. This method can be used by all iPhones and 3G enabled iPad's. The third method uses a WiFi hotspot, which is similar to the Cell Tower Triangulation but is only accurate to within a few hundred feet. All iPhone OS devices (iPhone, iPod Touch and iPad) support WiFi hotspot geolocation services.

Using Multitouch in an App

By default, Flash assumes that you are using a mouse to touch the screen. This actually works OK for many iPhone Apps and allows you to take the same code you have developed for other Flash movies and move it into your iPhone projects. That's OK, but what about those cool iPhone gestures (**New!**) you see people using to swipe content? Flash can do that, too.

For the iPhone, Flash supports the following Multitouch gestures:

- ◆ TransformGestureEvent.GESTURE_PAN
- ◆ TransformGestureEvent.GESTURE_ROTATE
- ◆ TransformGestureEvent.GESTURE_SWIPE
- ◆ TransformGestureEvent.GESTURE_ZOOM

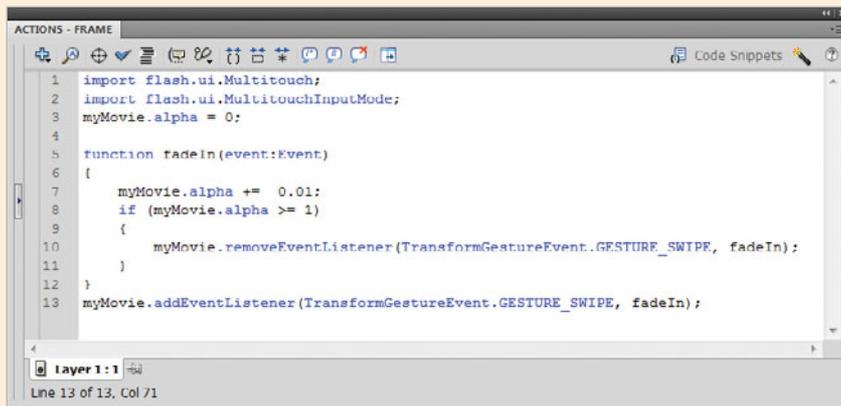
Each gesture is used as an interactive event in a Listener. This means that you can take your knowledge of Listeners and quickly apply it to iPhone Apps. The following example shows you how to add a swipe gesture that will trigger a movie clip to fade onto the screen.

Use the Multitouch Class

- 1 Create a new iPhone App, apply the appropriate development profiles and certificates, and then save the file with the name *iPhoneMultitouch fla*.
- 2 Draw a rectangle on the Stage, and then convert the rectangle into a Movie Clip with the name *myMovie*.
- 3 Open the **Actions** panel.
- 4 Add code to import the class libraries:

```
import flash.ui.Multitouch;  
import flash.ui.MultitouchInputMode;
```
- 5 Set the default Alpha value for the Movie Clip to 0:

```
myMovie.alpha = 0;
```
- 6 Add a listener and function that enable the Movie Clip to fade onto the screen (see illustration).
- 7 Publish and package your file into an iPhone App and test it on your iPhone.



```
1 import flash.ui.Multitouch;  
2 import flash.ui.MultitouchInputMode;  
3 myMovie.alpha = 0;  
4  
5 function fadeIn(event:Event)  
6 {  
7     myMovie.alpha += 0.01;  
8     if (myMovie.alpha >= 1)  
9     {  
10        myMovie.removeEventListener(TransformGestureEvent.GESTURE_SWIPE, fadeIn);  
11    }  
12 }  
13 myMovie.addEventListener(TransformGestureEvent.GESTURE_SWIPE, fadeIn);
```

Layer 1:1
Line 13 of 13, Col 71

Index

A

- Actions panel
 - breakpoints, setting, 423
 - viewing, 348
- ActionScript Editor, debugging with, 412-413
- ActionScript Server files, 14
- ActionScript Virtual Machine (AVM), 350
- ActionScripts, 10, 131, 347. *See also*
 - behaviors; components; conditional statements; debugging; dot syntax; methods; properties
- animation, creating, 288
- classes, 375-377
 - drawing shapes, 384
- clip events, 358-359
- code hints, 370-371
- code snippets, 372-373
- data, controlling, 380
- data types, 353
- DOM event model, 374
- for dynamic text, 186
- expressions, regular, 382
- external SWF files, loading, 451
- files, 14
- frame events, 357
- functions, using, 354
- motion tween as ActionScript, copying, 250, 259
- mouse events, 356
- Namespaces, 378-379
- new version, introducing, 350
- preferences, setting, 349
- printing with, 492
- settings, 3.0, 368-369
- similarities between 2.0 and 3.0, 367
- text, controlling, 383
- using, ActionScript 3.0, 366-367
 - video with, 334-335
 - warning preferences, 65
 - XML with E4X, 381
- activating masks, 301
- Adaptive color palette, 489
- Add Statement, 348
- Adjust Color effect, 246-247
- Adobe AIR, 482-483
 - create, 482, 510
 - install and start, 511
 - publish, 483
- Adobe After Effects, 340
- Adobe Bridge, 14-15
- Adobe ConnectNow, 523
- Adobe CS Live, 522
- Adobe Extension Manager, 524
- Adobe Fireworks. *See* Fireworks
- Adobe Flash Builder
 - create, ActionScript file, 506
 - open in, 506-507
- Adobe Flash Exchange Web site
 - commands, downloading, 475
 - Extend link for accessing, 4
 - resources on, 27
 - updates and support on, 26-27
- Adobe Flash Player. *See* Flash Player
- Adobe Flash Support Online Web site
 - templates on, 13
 - updates and support on, 26-27
- Adobe Flash Video. *See* Flash movies; FLV files
- Adobe FXG
 - import and export, 508-509
- Adobe Illustrator. *See* Illustrator
- Adobe Photoshop. *See* Photoshop
- Adobe Premiere for alpha masks, 342-343
- Adobe Soundbooth, 307, 323
- ADPCM compression, 320

- advanced colors. [See](#) colors
- advanced text options, 178-179
- Advertising templates, 13
- alignment
 - with guide layers, 41
 - HTML alignment, setting, 487
 - of paragraph text, 172
- alpha
 - instance alpha, modifying, 158, 160
 - masks, working with, 342-343
 - motion tween, adding to, classic, 268
 - publishing movies, setting on, 489
 - shape tween alpha, changing, 285
- anchor points, 81
 - arrow keys for moving, 116
 - corner points to curve points, converting, 116-117
 - curve points to corner points, converting, 116-117
 - deleting, 111
 - Pen tool using, 110
 - types of, 116-117
- AND conditional statement, 355
- animations. [See also](#) frame-by-frame animation; GIF files; motion tweening
 - anti-aliasing for, 182
 - buttons, adding to, 151
 - of filters, 247
 - masks, animating, 306
 - symbols, converting to, 139
 - sync sounds, adding, 309
- anti-aliasing text, 182
- API (application programming interface) and components, 427
- Apple Final Cut Pro, 340
 - alpha masks, working with, 342-343
- Arial font, 182
 - built-in Flash device fonts for, 185
- armature, 289-294
- Arrange command, 133
- arrow keys, 116
- assets in Library, 134-135
- attachMovie() method, 359
- attributes, list of, 465
- audience, determining, 8
- audio. [See](#) sounds
- Audio Video Interleaved (AVI) files. [See](#) AVI files

- authoring parameters, 430, 431
- Auto Close Bracket, 348-349, 371
- Auto Format, 348
- Auto Kerning, setting, 171
- AutoCAD DXF. [See](#) DXF files
- AVI files
 - exporting movies to, 502-503
- AvidXpress DV, 340
- AVM, 365

B

- Back button on Edit bar, 46
- backgrounds
 - attributes, 463
 - Property Inspector, changing color in, 16
- ball animation. [See](#) shape tweening
- bandwidth and debugging, 411
- batch processing video encoding, 340-341
- BBEdit, 351
- behaviors, 10-11
 - adding behaviors, 362
 - instance, changing symbol behavior of, 144-145
 - sounds with, 314-315
 - Start/Stop Dragging Movie clip Behaviors, 362
 - symbol behavior, changing, 144
 - video with, 334-335
 - warn on behavior symbol conversion, 65
- Behaviors panel, 362
- Bevel effect, 246-247
- bevel joins, 75
- Bézier handles, 81
 - Pen tool using, 110
- Bind tool, 289, 294
- bindings, 431
 - creating, 458-459
 - modifying, 460-461
- Bindings tab, Component Inspector, 429, 430. [See also](#) bindings
- bit depth options, 490
- bit rate for sounds, 321
- bitmaps
 - Break Apart for fill patterns, 211
 - center point of fill, changing, 212
 - compression, setting, 210
 - fills

- Break Apart for fill patterns, 211
 - center point, changing, 212
 - orientation, changing, 212-213
 - resizing, 212-213
- Find and Replace for, 190
- Fireworks files, importing, 204-205
- graphics, 194
- Illustrator, importing from, 196
- Magic Wand, editing with, 214-215
- orientation of fills, changing, 212-213
- outside editor, editing in, 208-209
- Photoshop import preferences, 198-199
- PNG bitmaps, importing, 204-205
- preferences, setting, 63
- replacing, 209
- resizing fills, 212-213
- Stage, importing bitmap to, 202
- support for, 195
- swapping, 203
- text, 182
- tracing bitmaps, 216
- blending
 - instance blend, modifying, 159
 - shape tween, setting for, 284
- Blur effect, 246-247
- BMP files, 193
- Bones tool, 289-294
 - properties, 292-293
- Boolean data, 353
- borders
 - for dynamic text, 186
 - for input text, 187
 - for Label components, 446
- Bound To dialog box, 461
- Break Apart, 163, 180-181
 - bitmap fill pattern, creating, 211
- breaking
 - instances, 155
 - setting breakpoints, 423
 - text fields, 180-181
- brightness
 - instances, modifying brightness of, 158
 - motion tween, adding to, classic, 268
- Bring to Front/Bring Forward, 133
- browsers
 - identifying playback browser, 9
 - printing from Flash Player, 492

- system requirements, 2
- building movies, 10-11
- Button component, adding, 442-443
- button objects, 351
- buttons. *See also* RadioButton
 - animation, adding, 151
 - creating, 150-151
 - enabling/disabling, 152
 - invisible buttons, creating, 153
 - mouse events, attaching, 356
 - window sizing buttons, 19

C

- calling functions, 302
- camera data file information, 484-485
- caps, 75
- case-sensitivity in ActionScripts, 347
- cell phones
 - content for PDA, creating, 512
 - deployment of Flash movies to, 515
- center of gradient fills, changing, 121
- centering text, 172
- characters. *See also* kerning
 - breaking text fields into, 180
 - for dynamic text, 186
 - for input text, 187
 - options, setting, 182
- circles, creating, 79
- classes
 - object-oriented programming, 364
 - working with ActionScript 3.0, 376-377
- Classic text, creating,
 - dynamic and input text, 186-187
 - static text, 164
- classic tween. *See* motion tweening
- Clear Keyframe function, classic, 264-265
- clip events, 358-359
 - attaching to movie clips, 359
- Clipboard
 - pasting images from, 207
 - preferences, setting, 63
- Close button, 19
- closing
 - documents, 28
 - History panel, 467
 - panels, 52

- code hints, 348
 - custom classes, 370-371
 - in debugging, 413
 - setting, 349
 - using, 370-371
- Code Snippets panel, 349, 372-373, 412
- ColdFusion, 448
- collapsing. *See* maximizing/minimizing
- Color panel
 - bitmap fill, selecting, 211
 - fill colors, changing, 97
 - gradients in, 105
 - for shape tween, 285
 - specifying colors in, 98
 - stroke color, changing, 98
 - text colors, changing, 169
- color palettes
 - importing, 99
 - for shape tween, 285
 - type options, 489
 - Web 216 color palette, 489
 - working with, 98-99
- Color Picker, 98-99
- colors. *See also* backgrounds; color palettes; GIF files; gradients; instances; PNG files; RGB colors; Swatches panel; syntax-coloring
 - advanced colors
 - motion tween, adding effects to, classic, 268
 - simultaneously modifying color and transparency, 160
 - attributes, 463
 - fill colors, changing, 96-97
 - Find and Replace for, 190
 - font colors, changing, 169
 - highlight color preferences, setting, 162
 - instances, modifying color of, 158-159
 - Kuler panel, 100-101
 - motion tween, adding effects to, classic, 268
 - shape tween colors, changing, 285
 - stroke color, changing, 98
 - TextArea component, adding, 440-441
 - tint color of instance, modifying, 158
 - values, creating custom colors with, 98
- column of vertical text, creating, 175-176
- column text, TLF, 174
- ComboBox component, adding, 434-435
- commands, 31
 - deleting names of, 474-475
 - help commands, 25
 - History panel, saving steps as command in, 474-475
 - for Onion Skin markers, 235
 - running command, 475
 - saving steps as, 474-475
- comments
 - in Actions panel, 348
 - with ActionScript Editor, 413
 - breakpoints, setting, 423
- compatibility of ActionScripts, 365
- Compiler Errors panel, 418-419
- Component Inspector, 429. *See also* components; parameters
 - authoring parameters, 430
 - Bindings tab, 430, 458
 - description of, 430
 - Schema tab, 430, 462, 462
- components, 10-11, 14, 427. *See also* bindings; DataGrid component; parameters; schemas
 - API (application programming interface) in, 427
 - attributes, modifying, 463
 - basic components, understanding, 428-429
 - Button component, adding, 442-443
 - categories of, 429
 - CheckBox component, adding, 436-437
 - ComboBox component, adding, 434-435
 - forms, building, 464
 - Label component, adding, 446-447
 - List component, adding, 444-445
 - Loader component, adding, 451
 - NumericStepper component, adding, 452-453
 - preloaders with, 451
 - ProgressBar component, adding, 454-455
 - RadioButton component, adding, 438-439
 - ScrollPane component, adding, 456-457
 - TextArea component, adding, 440-441
 - TextInput component, adding, 432-433
- compression
 - bitmap compression, setting, 210

- of Flash movies, 480
- for sounds, 320
- conditional statements, 360
 - using, 355
- connected lines, drawing, 130
- Consumer Devices, 13, 512
- context-sensitive help, 412
- ConvertTo Symbol dialog box, 139, 149
- Copy Frames command, 226-227
- Copy Steps button, 476
- copying
 - bitmap images, 207
 - files, 21
 - frame sequence, 43
 - keyframes, 43, 226
 - layers or folder contents, 38-39
 - motion tweens, 258-259
 - other programs, images from, 207
 - steps between documents, 476
- corner points to curve points, converting, 116-117
- CPU requirements, 9
- CS Live, 522
- CSS (Cascading Style Sheets), 191
- cue points
 - ActionScript, adding, 332-333
 - embedded, 340-341
- Current Script tag, 348
- curve points to corner points, converting, 116-117
- curves
 - drawing smooth curves, 130
 - line segments, creating, 112
 - Pen tool, drawing with, 112-113
 - S-curves, creating, 113
 - shapes, optimizing, 77
- customizing. *See also* colors
 - color palette, 489
 - keyboard shortcuts, 58
 - Tools panel, 50-51

D

- data binding. *See* bindings
- data components. *See* components
- data types, 353
- DataGrid component
 - adding, 448

- local data provider, using, 450
- options, modifying, 449
- Debugger, 416-420. *See also* properties; variables
 - ActionScript 2.0, 420
 - ActionScript 3.0, 416-417
 - breakpoints, setting, 424
 - compiler errors, 418-419
 - stepping through code, 425
- debugging, 409. *See also* Debugger
 - in Actions panel, 348
 - with ActionScript Editor, 412-413
 - bandwidth and, 411
 - breakpoints, 412
 - concepts of, 410-411
 - Flash movies, permitting debugging of, 480
 - Flash Player, 426
 - manual debugging, 426
 - with Movie Explorer, 414-415
 - planning and, 411
 - testing and, 411
- Deco tool, 109
- defaults
 - Snap Align defaults, saving, 95
 - Toolbar defaults, restoring, 51
- delay, setting, 349
- deleting. *See also* masks; motion tweening
 - anchor points, 111
 - Command menu, name from, 474
 - commands, names of, 474-475
 - files in dialog box, 22
 - font mapping, 183
 - frames, 43
 - keyboard shortcut sets, 58-59
 - keyframes, 224-225
 - layers/folders, 34-35
 - panel sets, 57
 - scenes, 45
 - shape tween, keyframes from, 278
- delivery computer system, identifying, 9
- deployment
 - Flash deployment kit, 478
 - to other devices, 515
- device fonts, setting, 184-185
- DICOM file information, 484-485
- dictionaries, using, 188
- Digital Juice, 308

- direction, text, TLF, 175
- display components. [See](#) components
- distorting shapes, 120
- Distribute to Layers command, 129
- dithering on publishing movies, 490
- docking/undocking panels, 54
- Document window, 7, 30-31
 - frame rates, changing, 262
- documents. [See also](#) layers; printing; projects
 - arranging multiple windows, 18
 - blank document, creating, 12
 - closing, 28
 - copying steps between documents, 476
 - creating
 - Adobe AIR, 482
 - existing documents, opening, 14-15
 - general preferences, setting, 60
 - languages for, 187
 - last saved version, reverting to, 20
 - launching, opening documents on, 5
 - mouse, resizing windows with, 19
 - moving document windows, 19
 - multi-language authoring, 187
 - projector document, creating, 493
 - properties
 - setting up, 10
 - working with, 16-17
 - recently opened documents, opening, 15
 - resizing document windows, 19
 - saving, 20
 - different name/location, saving with, 21
 - formats for, 22
 - multiple documents, 20
 - templates, saving as, 23
 - switching between, 18
 - templates
 - creating new document from, 13
 - saving document as, 23
- dot syntax
 - description of, 352
 - movie clips, applying to, 352
- Down state for buttons, 150
- downloading
 - commands, 475
 - 10-second rule, 504

- drag-and-drop
 - DataGrids, 448
 - keyframes, 227
 - Library items, 134-135
 - motion tween, keyframes in, 264
- drawing
 - Line tool for, 75
 - masks, 300
 - model options, 61
 - Oval tool, 78
 - Pencil tool for, 76-77
 - Polystar tool, 80
 - Rectangle tool, 78-79
 - rounded rectangle, 79
 - selection examples, 81
 - settings, changing, 130
- Drop Shadow effect, 246-247
- duplicating profiles, 498
- dynamic media, 10
- dynamic text. [See](#) text

E

- ease properties. [See also](#) motion tweening
 - shape tween, setting for, 284
- ECMAScript, 348
- Edit bar, 7
 - scenes, selecting, 44
 - showing/hiding, 46
- Edit in New Window mode, 140
- Edit In Place mode, 140
- Edit Snapping dialog box, 94-95
- editable parameters, 430
- editing. [See also](#) bitmaps; outside editors; symbols
 - keyframes, 226-227
 - Library, items in, 136-137
 - masks, 302
 - Onion Skins, 233
 - profile properties, 495
 - scene for editing, selecting, 47
 - sounds, 322
 - with Video Encoder, 340-341
- effects, sound, 319
- e-mail
 - sending a document, 521
- embedding. [See also](#) fonts
 - audio files, 309

- data providers, 450
 - empty keyframe, creating, 220-221
- Enable Simple Buttons command, 152
- Enhanced Metafile (EMF)
 - support for, 195
- EPS files, 193
 - importing options, 200
- equals (=) statement, 355
- event handlers
 - clip events, 358-359
 - frame events, 357
- events. *See also* ActionScripts; event handlers
 - mouse events, 356
- existing documents, opening, 14-15
- expanding/collapsing. *See* maximizing/minimizing
- exporting. *See* importing/exporting
- Extend link, 4
- Eye icon, 36
- Eyedropper tool
 - for bitmap fills, 211
 - fills, editing, 104
 - strokes, editing, 104

F

- file formats
 - exporting movies to different formats, 502-503
 - import file formats, 195
 - list of existing documents, opening, 14-15
- files
 - debugging, file names and, 410
 - deleting, 22
 - text files, loading, 164
- fills. *See also* bitmaps; gradients
 - Selection tool, selecting with, 84
 - colors, changing, 96-97
 - Eyedropper tool, editing with, 104
 - locking gradients, 106
 - Paint Bucket, editing with, 103
 - text fills, 169
- filter effects, applying, 246-247
- Find and Replace, 190
 - for active script, 348
- Fireworks
 - importing files, 204-205

- pasting images from, 207
- FLA files. *See* documents
- Flash, 1
- Adobe Flash Builder
 - create, ActionScript file, 506
 - open in, 506-507
- Flash deployment kit, 478
- Flash File for mobile devices, 512
- Flash Live Video files. *See* FLV files
- Flash movies, 14 *See also* debugging; publishing movies
 - Bandwidth Profiler, working with, 504
 - different formats, exporting movie to, 502-503
 - exporting
 - different formats, exporting movies to, 502-503
 - video clips into FLV format, exporting, 346
 - Flash Player, optimizing for, 480
 - For loops for pausing, 361
 - planning, 8
 - previewing, 499
 - progressive downloads, 346
 - search engine searchability, improving, 17
 - security for local/network playback, 481
 - testing, 501
 - video-editing applications, exporting files from, 340-341
- Flash Player, 2
 - debugging, 426
 - component, 338-339
 - frame-by-frame animation, testing, 231
 - installation, 3
 - magnification, changing, 499
 - masking and, 298
 - optimizing movies for, 480
 - printing from, 492
 - support for, 195
 - Version Detection, 487
- Flash Project files, 14
- Flash Remoting, 448
- Flash Video Import Wizard, 325
- Flash window, 30-31
- FlashType, 182
- flipping instances, 157
- flowcharts, developing, 8

- FLV files, 10, 327. *See also* Flash movies
 - exporting directly to, 346
 - Property Inspector, modifying video clips in, 346
 - FLV (Flash Video) QuickTime Export plug-in, 340-341
 - FLVPlayback component, 338-339
 - folders. *See also* layers
 - Library assets, creating folders for, 12
 - overlapped root folder warning, 65
 - renaming (*See* naming/renaming)
 - Font Mapping dialog box, 183
 - fonts
 - ActionScript text, setting for, 349
 - anti-aliasing, 182
 - attributes, 463
 - built-in Flash device fonts, 185
 - colors, changing, 168-169
 - deleting font mapping, 183
 - device fonts, setting, 184-185
 - embedded fonts, 184
 - warning, 64-65
 - Find and Replace for, 190
 - mapping, 183
 - missing font information, 184
 - preferences, setting, 62
 - size, changing, 169
 - substitute fonts, selecting, 183
 - symbols to embed font, creating, 136-137
 - TextArea component, adding, 440-441
 - type, changing, 168
 - warn on missing fonts, 64
 - For loops, 360
 - working with, 361
 - foreign languages. *See* languages
 - forms
 - components for building, 464
 - 45 degree angles, creating, 111
 - frame-based selection, 223
 - frame-by-frame animation, 217. *See also* Onion Skinning
 - Controller, playing with, 230
 - description of, 218-219
 - exporting to Flash movie, 230
 - Flash Player, testing in, 231
 - keyframes, 217
 - clearing, 224
 - converting frames to, 222
 - copying, 226
 - creating, 220-221
 - deleting, 224-225
 - drag-and-drop, 227
 - editing, 226-227
 - empty keyframe, creating, 220-221
 - pasting, 226
 - removing content from, 220
 - selecting frames, 223
 - span of keyframe, increasing, 221
 - multiple frames, editing, 236
 - playing animation, 230-231
 - selecting frames, 223
 - simple animation, creating, 228-229
 - frame events, 357
 - frame rate, 16
 - motion tween, changing in, 262
 - Property Inspector, changing in, 16
 - frames. *See also* keyframes; scenes
 - copying frame sequence, 43
 - deleting, 43
 - inserting, 43
 - moving frame sequence, 43
 - selecting, 42
 - warn on inserting frames when importing content, 64
 - Windows, setting in, 68
 - Free Transform tool
 - distorting shapes in, 120
 - instances, working on, 156-157
 - scale of object, changing, 118
 - for text, 181
 - Free Transform tool
 - envelope of shape, changing, 120
 - rotating object with, 119
 - skewing object with, 119
 - functions, using, 354
 - FXG file format
 - importing and exporting, 508-509
- ## G
- general preferences, setting, 60-61
 - Generator content, warn on reading, 64
 - geometric forms for shape tweening, 276
 - GIF files, 193
 - lossless compression with, 210

- publishing movies, specifying options when, 488-489
- support for, 195
- Global Transform, 3D, 122-123
- Glow effect, 246-247
- Gradient Bevel effect, 246-247
- Gradient Transform tool, 121
 - for bitmap fills, 212-213
- gradients
 - locking, 106
 - multi-color gradient, creating, 105
 - publishing movies, settings on, 490
 - quality, setting, 63
 - transforming gradient fills, 121
- graphic symbols, 148-149
- graphics, 73. *See also* bitmaps; drawing; importing/exporting; layers; symbols; vector graphics
 - bitmap graphics, 194
 - filter effects, applying, 246-247
 - grouping/ungrouping, 132
 - pasting objects between layers, 126-127
 - symbols, converting to, 149
 - vector graphics, 194
 - warning preferences, setting, 65
- greater than (>) statement, 355
- greater than or equal to (>=) statement, 355
- grids and guides. *See* motion tweening
 - creating guides, 91
 - guide layers, 40-41
 - locking/unlocking, 91
 - modifying, 92-93
 - showing/hiding, 90
 - snap to grid, locking/unlocking, 91
 - turning on/off, 91
- grouping/ungrouping, 131
 - artwork, 132
 - Illustrator import preferences, 197
 - panels, 55
 - Photoshop import preferences, 198-199
 - stacking order, changing, 133
- guides. *See* grids and guides

H

- Hand tool, 47, 88
- hardware requirements, 9
- help, 24-25

- in Actions panel, 348
 - commands, 25
 - context-sensitive help, 412
- Helvetica font, 182
 - built-in Flash device fonts for, 185
- hidden layers. *See* publishing movies
- hiding. *See* showing/hiding
- highlight color preferences, setting, 162
- hints. *See also* code hints
 - for shape tween, 286-287
 - Stroke hinting, 75
- History file information, 484-485
- History panel, 465
 - clearing, 467
 - command, saving steps as, 474-475
 - copying steps between documents, 476
 - description of, 466
 - object level undo, working with, 468-469
 - opening/closing, 467
 - Options button, 466
 - repeating previous steps with, 466, 470
 - replaying steps with, 472-473
 - undoing/redoing with, 466, 470-471
 - view, changing, 473
- horizontal lines, creating, 111
- HTML (Hypertext Markup Language). *See also* publishing movies
 - CSS (Cascading Style Sheets), 191
 - keyboard shortcuts as HTML, exporting, 59
 - text rendered as, 186
- HTTP/HTTPS security, 481

I

- IK animations and bones, 289-293
- Illustrator, 193
 - converting layers options, 201
 - exporting Flash movies to, 502-503
 - file information, 484-485
 - import preferences, setting, 196-197
 - options for importing files, 200-201
 - Pen tool, 110
 - support for, 195
- Import dialog box, 206
- Import Script, 412
- ImportTo Stage method, 202

- importing/exporting. *See also* Flash movies; publishing movies; video; specific file formats
 - color palettes, importing, 98-99
 - different formats, exporting movie to, 502-503
 - file formats for, 195
 - Fireworks PNG files, 204-205
 - FLV format, exporting directly to, 346
 - frame-by-frame animation , exporting, 230
 - keyboard shortcuts as HTML, exporting, 59
 - media elements, 10
 - multiple files, importing, 206
 - profiles, 496-497
 - scripts, 412
 - sequence of files, importing, 206
 - sounds, 308-309
 - warning preferences, setting, 64-65
- incompatibility warning preferences, 65
- indents for text, setting, 173
- inequality statement, 355
- Info panel, 88
- inheritance, 377
- Ink Bottle tool, 102
- iPhone, 385-408
 - Accelerometer, in App, 394
 - App ID, creating, 389
 - Camera Roll, saving images, 395
 - Device ID, 397
 - Developer, becoming, 386-387
 - Geolocation, in App, 407
 - limits of Flash, 396
 - Multitouch, in App, 408
 - Profiles
 - Developer Provisioning, 390-391
 - Ad Hoc Distribution
 - Publishing/Packaging
 - Ad Hoc, 399
 - iPhone App, 392-393
 - iTunes App Store, 400-401
 - iTunes Connect, 402-405
 - registering an iPhone or testing device, 388
 - Screen Orientation, in App, 406
- input method, setting, 62
- input text. *See* text

- IPTC file information, 484-485
- inserting file information, 484-485
- Insert Target Path, 348
- installation
 - preparation for, 2
 - standard application install, 3
- instances
 - advanced color options, 160
 - behavior, changing, 144-145
 - blend, modifying, 159
 - breaking, 155
 - brightness, modifying, 158
 - for dynamic text, 186
 - flipping, 157
 - for graphic symbols, 148-149
 - for input text, 187
 - movie clip instances, 154
 - multiple instances, working on, 156
 - properties, changing, 431
 - rotation, modifying, 156-157
 - scale, modifying, 146-157
 - simultaneously modifying color and transparency, 160
 - Stage, placing on, 142
 - swapping, 146-147
 - tint color, modifying, 158
 - transparency, modifying, 159, 160
- Integrated Script window, 413
- interlacing on publishing movies, 490
- Inverse Kinematics, 289-294
- invisible buttons, creating, 153

J

- Java recordsets, 448
- JavaScript, 14
 - commands, running, 475
 - History panel steps, viewing, 473
- joins, 75
- JPEG files, 10,193
 - publishing movies, specifying options when, 492
 - Quality slider, dragging, 481
 - support for, 195
- JSFL language, 51
- justifying text, 172

K

- kerning
 - adjusting, 170-171
 - Auto Kerning, setting, 171
 - preferences, setting, 62
- keyboard controlling radio buttons, 438
- keyboard shortcuts, 31
 - adding, 59
 - deleting, 58-59
 - HTML, exporting as, 59
 - open path, ending, 110
 - sets, creating, 58
 - Toolbar, identifying for, 51
- keyframes, 42. *See also* frame-by-frame animation; motion tweening; shape tweening
 - copying, 43, 226
 - moving, 43
- Kuler panel, 100-101

L

- Label component, adding, 446-447
- labels
 - Label component, adding, 446-447
- landscape orientation
 - Macintosh, setting in, 66
 - Windows, setting in, 68
- languages
 - ActionScript language, setting, 349
 - character options, setting, 182
 - multi-language authoring, 187
- Lasso tool
 - general preferences, setting, 61
 - with Magic Wand, 215
 - in polygon mode, 85
 - selecting with, 85
- launching
 - Flash, 4-5
 - preferences on launch, setting, 60
- Layer Properties dialog box, 40, 303
- layers, 40. *See also* masks; motion tweening
 - copying layers or folder contents, 38-39
 - deleting layers/folders, 34-35
 - distributing graphics to, 129
 - expanding/collapsing folders, 38
 - guide layers, 40-41
 - height, changing, 40
 - Illustrator import preferences, 197
 - inactive layers, editing object on, 128
 - layer folder, creating, 34
 - locking/unlocking, 36, 40
 - moving layers/folders, 38
 - multiple layers, selecting, 35
 - naming/renaming, 34, 35, 40
 - new layers/folders, creating, 34
 - outlines, viewing contents as, 36-37
 - pasting objects between, 126-127
 - Photoshop import preferences, 198-199
 - pose, 289
 - properties, changing, 40
 - renaming, 34, 35, 40
 - selecting, 34-35
 - separate layers, placing objects on, 129
 - showing/hiding, 36, 40
 - types of, 40
- layout
 - designing, 8-9
 - Windows, setting in, 68
- leaf of Tree component, 426
- left-aligning text, 172
- less than (<) statement, 355
- less than or equal to (<=) statement, 355
- Library, 123. *See also* importing/exporting; sounds; symbols; video
 - accessing options, 137
 - bitmaps, importing, 202-203
 - items, 350
 - opening, 134-135
 - organizing item, 136
 - other libraries, opening, 134-135
 - sample common libraries, 134-135
 - sounds, importing, 308-309
- line segments
 - Selection tool, converting with, 115
 - Pen tool, creating with, 112
- line spacing, setting, 173
- Line tool, 75
- linear gradients, 105
- lines. *See also* line segments
 - connected lines, drawing, 130
 - 45 degree angles, creating, 111
 - horizontal lines, creating, 111
 - recognize lines option, 130

lines (continued)

- vertical lines, creating, 111
 - width, changing, 75
 - linked layers. *See* masks
 - linking Library assets, 137
 - linking text, TLF, 166-167
 - List component, adding, 444-445
 - Loader component, adding, 451
 - Lock Fill feature, 106
 - Lock icon, 36
 - locking/unlocking
 - gradients, 106
 - grids and guides, 91
 - layers, 36, 40
 - masks, 302
 - loops
 - creating, 360
 - exceptions, working with, 360
 - for graphic symbols, 148
 - sounds, looping, 319
 - lossless compression, 210
 - lossy compression, 210
- ## M
- Macintosh computers
 - keyboard commands on, 1
 - launching Flash, 5
 - margins, setting, 66-67
 - page setup, working with, 66-67
 - printing documents, 70-71
 - projector, creating, 493
 - system requirements for, 2
 - Magic Wand bitmaps, editing, 214-215
 - magnification
 - Flash Player, changing in, 499
 - Magnifying Glass, 86-87
 - Magnifying Glass, 86-87
 - Main toolbar, 30-31
 - showing/hiding, 48
 - mapping MIDI device sounds, 514-515
 - margins
 - attributes, 463
 - Macintosh, setting in, 66-67
 - text margins, setting, 173
 - Windows, setting in, 68
 - markers. *See* Onion Skinning
 - mask layers, 40

- masked layers, 40
- masks, 295
 - activating, 301
 - adding layers to, 304
 - alpha masks, working with, 342-343
 - animating, 306
 - creating mask layers, 298-299
 - deleting
 - linked layers, 305
 - mask layers, 303
 - description of, 296-297
 - drawing masks, 300
 - editing, 302
 - linked layers
 - activating masks, 301
 - deleting, 305
 - order, changing, 304
 - Lock Layer column, 302
- master symbols. *See* symbols
- Maximize button, 19
- maximizing/minimizing
 - in Actions panel, 348
 - with ActionScript Editor, 413
 - layer folders, 38
 - panels, 52-53
 - Tools panel, 50
- media components, 429
- Media Controller
 - for frame-by-frame animation, 230
- media elements, 10
- menus, 31
- Merge Drawing, 61, 73
 - converting to Object Drawing shape, 74
- Metadata, 351
 - file information, 484-485
- methods, 364, 350. *See also* ActionScripts; parameters
 - applying, 351
- MIDI device sounds, mapping, 514-515
- Minimize button, 19
- minimizing. *See* maximizing/minimizing
- miter joins, 75
- Mobile SWF file information, 484-485
- mobile devices. *See* cell phones; PDAs (personal data assistants)
- monitor resolution requirements, 9
- Motion Editor, 244-249

- motion tweening
 - adding color effects and filters, 246-247
 - adding frames to tween, 260
 - adjusting, motion, 243
 - creating,
 - classic, 251
 - motion, 242
 - color effects, adding, classic, 268
 - copying tween, 258-259
 - deleting
 - frames from tween, 260-261
 - keyframes, classic, 264-265
 - ease in/out properties
 - customizing, classic, 272
 - setting, classic, 257
 - edit path, motion, 249
 - frame properties, classic, 253
 - frame rate, changing, 262
 - instance properties, 253
 - keyframes
 - adding, 255, 264
 - deleting, classic, 264-265
 - moving, 255
 - work with, motion, 248
 - length of tween, changing, 260-261
 - Motion Editor, 244-249
 - motion guides
 - animating along, classic, 270
 - creating, classic, 269
 - orienting objects along, classic, 271
 - pasting tween, 258-259
 - presets, motion tween, 240-241
 - properties
 - ease properties, setting, classic, 257
 - frame and instance properties compared, 253
 - pasting properties, 259
 - setting, classic, 256-257
 - reversing frames, 263
 - rotation, changing, classic, 266-267
 - scale
 - changing, classic, 266
 - rotate and scale, working with, 119
 - simple motion tween, creating, classic, 254-255
 - smoother animation, adding frames for, 261
 - target object, 237

- understanding,
 - classic, 238-239, 252
 - differences, 238
 - motion, 237, 238-239
 - working with, classic, 252
- mouse
 - events, 356
 - resizing windows with, 19
- movie clips, 154. *See also* components
 - clip events, 358-359
 - controlling video through, 336-337
 - dot syntax, applying, 352
 - Illustrator import preferences, 196-197
 - 9-slice scaling, using, 161
 - Photoshop import preferences, 198-199
 - placing video in, 331
 - scaling, 161
 - ScrollPane component for, 456-457
 - Start/Stop Dragging Movie clip Behaviors, 362
 - symbols, 154
 - Timeline, referencing to, 335
- Movie Explorer, debugging with, 414-415
- moving
 - document windows, 19
 - files, 21
 - frame sequence, 43
 - keyframes, 43
 - layers/folders, 38
 - Tools panel, 50
- MPG/MPEG files, 327
- MP3 files, 10
 - compression, 321
 - streaming MP3 file, loading, 316-317
- multi-language authoring, 187
- multi-purposing, 478
- music. *See* sounds

N

- named anchor on scene, setting, 61
- naming/renaming
 - Command menu, deleting name from, 474
 - command names, deleting, 474-475
 - debugging, file names and, 410
 - folders, 21
 - keyboard shortcut sets, 58

naming/renaming (continued)

- layers, 34, 35, 40
- with Movie Explorer, 414
- panel sets, 57
- scenes, 44-45
- smart conventions, using, 410-411
- navigating
 - in help topics, 24
 - planning for, 10-11
- .Net recordsets, 448
- nodes of Tree component, 426
- normal layers, 40
- Normal selection mode, 81
- number data, 353
- NumericStepper component, adding, 452-453

O

- object data, 353
- objects
 - working with ActionScript 3.0, 376-377
- Object Drawing, 61, 73
 - warning preferences, 65
 - working with, 74
- objects, 364. *See also* ActionScripts; properties
 - tracking per-object changes, 468-469
- On2 VP6 codec, 346
- onClipEvent(), 358, 359
- Onion Skinning, 217
 - activating, 232
 - manually setting markers, 234
 - Modify Onion Markers Window, setting markers in, 235
 - outlines, using, 233
- online activation requirement, 3
- opaque setting for publishing movies, 489
- Open dialog box, 21
- open path, Pen tool for, 110
- opening
 - existing documents, 14-15
 - History panel, 467
 - launching, opening documents on, 5
 - panels, 52
 - Scene panel, 44
- Options menu, 348
- OR conditional statement, 355

- orientation. *See also* Landscape orientation; Portrait orientation
 - bitmap fill orientation, changing, 212-213
 - text orientation, 176-177
- Outline icon, 36
- outlines of layer contents, 36-37
- outside editors
 - ActionScript code, editing, 351
 - bitmaps, editing, 208-209
- Oval tool, 78
- ovals, creating, 61, 70
- Over state for buttons, 150
- overlapped root folder warning, 65

P

- page setup
 - in Macintosh, 66-67
 - Windows, working in, 68-69
- Page Setup dialog box
 - in Macintosh, 66-67
 - in Windows, 68-69
- page size settings, 68
- Paint Behind mode, 107
- Paint Bucket
 - fills, editing, 103
 - fills with, 98
 - sensitivity, setting, 103
- Paint Fills mode, 107
- Paint Inside mode, 107
- Paint Normal mode, 107
- Paint Selection mode, 107
- Paintbrush tool, 98, 107
- panel sets
 - creating, 56
 - deleting, 57
 - displaying, 56
 - renaming, 57
- Panel windows, 6
- panels, 30
 - auto-collapsing, 60
 - docking/undocking, 54
 - expanding/collapsing, 52
 - grouping/ungrouping, 55
 - opening/closing, 52
- Panels Options menu, 53
- paper size settings, 66-67

- parameters
 - authoring parameters, 430, 431
 - editing, 430-431
 - video, 328-329
 - working with, 431
- Parameters tab, Component Inspector, 429
- parameters to functions, passing, 302
- password parameters, 430
- Paste Frames command, 226-227
- Paste In Center, 126-127
- Paste In Place, 126-127
- Pasteboard. [See](#) Work Area
- pasting
 - Clipboard, pasting images in, 207
 - keyframes, 226
 - layers, objects between, 126-127
 - motion tweens, 258-259
- paths
 - bitmap, preserving paths on importing, 208
 - in debugging, 410
 - Illustrator import preferences, 197
- pausing movie with For loops, 361
- PDA's (personal data assistants)
 - content for PDA, creating, 512
 - deployment of Flash movies to, 515
- PDF files, 193
 - importing options, 200
 - Macintosh documents, printing, 71
- Pen tool,
 - closed path, creating, 111
 - curves, drawing, 112-113
 - deleting anchor points with, 111
 - description of, 111
 - open path, creating, 110
 - options, 130
 - S-curves, creating, 113
 - selection with, 81
 - Show Precise Cursors option, 112
- Pencil tool,
 - in Smooth mode, 76-77
 - in Straighten mode, 76
- perspective angle, 3D, 124
- photo compression, 210
- Photo Slideshows, 13
- Photoshop, 193
 - converting layers options, 201
 - editing in, 207
 - file import preferences, setting, 198-199
 - options for importing files, 200-201
 - pasting images from, 207
 - support for, 195
- Pin Script, 348
- pinning multiple scripts, 413
- pixels, 194
 - document, setting for, 16-17
- planning
 - debugging and, 411
 - movies, 8
 - project plans, creation, 8-9
- Play Once for graphic symbols, 148
- playhead, 42
- playing
 - frame-by-frame animation, 230-231
- PNG files, 193
 - importing, 204-205
 - lossless compression with, 210
 - publishing movies, specifying options when, 490-491
 - support for, 195
- polygons
 - Lasso tool for, 85
 - with Polystar tool, 80
- polymorphism, 377
- Polystar tool, 80
- portrait orientation
 - Macintosh, setting in, 66
 - Windows, setting in, 68
- pose layer, 289
- preferences
 - ActionScript preferences, setting, 349
 - Clipboard preferences, setting, 63
 - general preferences, setting, 60-61
 - text preferences, setting, 62
 - warning preferences, setting, 64-65
- Preferences dialog box. [See](#) drawing
- preloaders, 451
 - ProgressBar component as, 454
- previewing, 11
 - Flash movies, 499
 - Macintosh documents, 71
- Primitive mode, 61
- primitive rectangles/ovals, creating, 61, 79
- Primitive shapes, 78

- printers
 - Macintosh, setting properties in, 67
 - Windows, setting properties in, 69
- printing. *See also* margins
 - with ActionScripts, 492
 - from Flash Player, 492
 - general preferences, setting, 61
 - help topics, 25
 - Macintosh documents, 70-71
 - Windows documents, 72
- product updates. *See* updates
- profiles. *See* publishing movies
- ProgressBar component, adding, 454-455
- progressive downloads, 346
- Project panel, 516-519
- projector document, creating, 493
- projects
 - creating and managing, 518-519
 - general preferences, setting, 61, 519
 - plans, creating, 8-9
 - testing, 520
- properties, 350. *See also* documents;
motion tweening; parameters
 - applying, 351
 - instance properties, changing, 431
 - layer properties, changing, 40
 - of movie clip symbols, 154
 - shape tween properties, setting, 284
 - viewing, 423
- Property Inspector, 6, 30-31. *See also*
 - behaviors; motion tweening
 - background color, changing, 16
 - bitmaps, swapping, 203
 - Color list arrow, 158
 - fill colors, changing, 96-97
 - Flash FLV video clips, modifying, 346
 - frame rate, changing, 16
 - kerning, adjusting, 170
 - motion tweens, adjusting, classic, 256
 - stroke color, changing, 98
 - text, rotating, 176
- Publish Preview command, 499
- Publish Settings dialog box, 346, 477
 - security options, 481
- publishing movies, 11, 477
 - Adobe AIR, 482-483
 - Bandwidth Profiler, working with, 504
 - considerations for, 478

- duplicating profiles, 498
- editing profile properties, 495
- GIF options, specifying, 488-489
- hidden layers
 - exporting, 481
 - viewing, 37
- HTML files, 11
 - options, specifying, 486-487
- importing/exporting
 - different formats, exporting movie to, 502-503
 - hidden layers, exporting, 481
 - profiles, 496-497
 - protecting from import, 480
- iPhone
 - Ad Hoc, 399
 - iPhone App, 392-393
 - iTunes App Store, 400-401
 - iTunes Connect, 402-405
- JPEG options, specifying, 492
- modifying publish settings, 479
- options, specifying, 480-481
- Photoshop import preferences, 199
- PNG options, specifying, 490-491
- previewing movies, 499
- profiles
 - creating, 494
 - duplicating, 498
 - editing properties of, 495
 - importing/exporting, 496-497
- projector document, creating, 493
- sound documents, 320-321
- testing movies, 501
- purpose of project, determining, 8
- PushButton type, 151

Q

- QuickTime Export Settings dialog box, 344-345
- QuickTime Image support, 195
- QuickTime/QuickTime Pro
 - exporting movies to, 344-345, 502-503
 - FLV Export Plug-in support, 340
 - version requirements, 2
 - video file support, 327
- quitting Flash, 28

R

- radial gradients, 105
- RadioButton, 151
 - component, adding, 438-439
- RAM requirements, 9
- raw data file information, 484-485
- raw sounds, 321
- readability, anti-aliasing for, 182
- recordsets, DataGrid component for, 448
- Rectangle Settings dialog box, 79
- Rectangle tool, 78-79
 - rounded rectangle, drawing, 79
- rectangles
 - Selection tool, selecting with, 84
 - primitive rectangles, creating, 61, 79
- redoing. *See* undoing/redoing
- registration point, creating, 139
- renaming. *See* naming/renaming
- repeating previous steps with History panel, 466, 470
- Replace. *See* Find and Replace
- replaying steps with History panel, 466, 467-470
- resizing
 - bitmap fills, 212-213
 - bitmap graphics, 194
 - document windows, 19
 - fonts, 168
 - gradient fills, 121
 - text boxes, 164
 - Timeline display, 32
- Restore Down button, 19
- Reverse Frames command, 263
- RGB colors
 - instances, modifying colors of, 160
 - values, entering, 98
- right-aligning text, 172
- Rotate Text button, 177
- rotation
 - with Free Transform tool, 119
 - gradient fills, 121
 - instance, modifying rotation of, 156-157
 - motion tween, changing during, classic, 266-267
- Rotation tool, 3D, 122-123
- of text, 175-177
- working with, 119

- round joins, 75
- rounded rectangle, drawing, 79
- rulers. *See also* grids and guides
 - showing/hiding, 89
 - unit of measure, specifying, 17, 89

S

- S-curves, creating, 113
- sample rate for sounds, 321
- sans serif fonts, 182
 - built-in Flash device fonts for, 185
- Save As dialog box
 - formats for files, 22
 - saving Flash FLA file, 20-21
 - saving Flash uncompressed XFL file, 21
- saving. *See also* documents
 - Snap Align defaults, 95
 - undo, saving documents after, 471
 - warning preferences, setting, 64
- scale. *See also* motion tweening
 - Free Transform tool for changing, 118
 - instance, modifying scale of, 156-157
 - Macintosh, setting in, 66-67
 - movie clips, 161
 - 9-slice scaling, using, 161
 - Windows, setting in, 68
 - working with, 119
- Scene button, 46
- Scene panel, 44
- scenes
 - adding scenes, 44
 - deleting, 45
 - edit, selecting scene to, 47
 - named anchor on scene, setting, 61
 - naming/renaming, 44-45
 - playback, 45
 - reordering, 45
 - selecting, 44-45
 - testing, 45
- Schema tab, Component Inspector, 429, 430, 462, 431. *See also* schemas
- schemas
 - description of, 462
 - modifying, 462
- schematic flowcharts, 8
- script, developing, 8
- Script Assist, 412, 448

- Script Navigator, 348, 413
- Script pane, 348
- scroll bars, 456-457
- scrolling
 - DataGrid component with, 448
 - text, creating, 187
- ScrollPane component, adding, 456-457
- scrubbing, 230
- search engines, 17
- searching
 - with Find and Replace, 190
 - for help information, 25
 - with spell-checking, 188
- security for local/network playback, 481
- Selection tool, 81
 - connected stroke segments, selecting, 83
 - fills, selecting, 84
 - line segments, converting, 115
 - multiple stroke segments, selecting, 83
 - rectangles, selecting, 84
 - shapes, modifying, 114
 - strokes, selecting, 82-83
- selecting
 - with Selection tool, 82-83
 - contact-sensitive selection, setting, 61
 - frames, 42, 223
 - with Lasso tool, 85
 - layers, 34-35
 - scenes, 44-45
 - span based selection, 61
 - text, 172
- Send to Back/Send Backward, 133
- sending a document, e-mail, 521
- serial number for installation, 3
- serif fonts, built-in Flash device fonts for, 185
- shape hints, 286-287
- shape tweening, 273
 - alpha, changing, 285
 - ball animation
 - combining tweening and movement in, 282-283
 - creating, 280-281
 - blending properties, setting, 284
 - colors, changing, 285
 - combining tweening and movement in ball animation, 282-283
 - creating shape tween form, 276-277
 - deleting keyframes, 278-279
 - hints, setting, 286-287
 - keyframes
 - adding, 278
 - deleting, 278-279
 - properties, setting, 284
 - working with, 274-275
- shapes. *See also* masks; shape tweening
 - Selection tool, modifying with, 114
 - breaking text fields into, 180-181
 - distorting shapes, 120
 - envelope of shape, changing, 120
 - optimizing, 77
 - recognize shapes option, 130
 - with Rectangle tool, 78-79
 - rough shapes, converting, 76
 - Stage, creating shapes on, 448
 - Subselection tool, modifying with, 116-117
- shared audio libraries, working with, 312-313
- Sharing My Screen, 523
- shortcuts. *See also* keyboard shortcuts
 - Macintosh, shortcut to Flash on, 5
- Show Precise Cursors, 112
- showing/hiding
 - Edit bar, 46
 - grids and guides, 90
 - layers, 36, 40
 - Main toolbar, 48
 - rulers, 89
 - Tools panel, 50
 - Welcome screen, 4
- Single Frame for graphic symbols, 148
- site maps, 8
- size reports, generating, 480
- skewing with Free Transform tool, 119
- Skinning component, 326
- skins, 326
- smooth curves. *See* curves
- Smooth mode, Pencil tool in, 76-77
- Snap Align, 94-95
- snap to grid, locking/unlocking, 91
- Snap to Object command, 95
- Sorensen codec, 346
- sound card requirements, 9
- Soundbooth, 307, 323

- sounds, 307
 - with ActionScript 3.0, 314-315
 - with ActionScript behaviors, 314-315
 - with behaviors, 314-315
 - bit rate for, 321
 - editing, 322, 323
 - effects, 319
 - embedded audio files, 309
 - external library, adding audio with, 310-311
 - Find and Replace for, 190
 - importing, 308-309
 - Library
 - external library, adding audio with, 310-311
 - importing sounds to, 308-309
 - shared audio libraries, working with, 312-313
 - Timeline, adding sounds to, 310-311
 - looping sounds, 319
 - MIDI device sounds, mapping, 514-515
 - MP3 file, loading, 316-317
 - publishing sound documents, 320-321
 - quality, changing, 321
 - shared audio libraries, working with, 312-313
 - Soundbooth, 323
 - stopping sounds
 - all sounds, 317
 - streaming MP3 file, loading, 316-317
 - sync sounds
 - animation, adding to, 309
 - Timeline, adding to, 318
 - Timeline
 - animation, adding sounds to, 309
 - Library, transferring file from, 310-311
 - sync sounds, adding, 318
 - timing of, 322
 - types, 307
 - zooming on, 322
- span-based selection, 61, 223
- Spanish text, displaying, 187
- Spray Brush tool, 108
- speech files, 321
- spell-checking, 188-189
- Spelling Setup dialog box, 188-189
- squares, creating, 79
- stacking order, changing, 133
- Stage, 7, 29, 30-31. *See also* frames; importing/exporting; layers; sounds; Work Area
 - bitmap to Stage, importing, 202
 - components, adding, 429
 - dimensions, setting, 17
 - Hand tool, moving with, 47, 88
 - shapes directly on stage, creating, 448
 - Snap Align, 94-95
 - video files on, 330
 - view size, changing, 47
 - zooming specific area on, 86-87
- Start menu, 15
- Start/Stop Dragging Movie clip Behaviors, 362
- startup preferences, customizing, 4
- static media, 10
- static text. *See* text, Classic text, and TLF text
- stepping through code, 425
- steps. *See* History panel
- stopping sounds. *See* sounds
- Straighten mode, Pencil tool in, 76
- streaming MP3 file, loading, 316-317
- streaming video. *See* video
- string data, 353
- Stroke hinting, 75
- strokes. *See also* lines
 - colors, changing, 98
 - Eyedropper tool, editing with, 104
 - Ink Bottle tool, editing with, 102
- Subselection tool, 81
 - anchor points and vectors, editing, 110
 - corner points to curve points, converting, 116-117
 - curve points to corner points, converting, 116-117
 - shapes, modifying, 116-117
- supplies for Macintosh printing, 71
- Swap dialog box, 146-147
- Swap Symbol feature, 146-147
- swapping bitmaps, 203
- Swatches panel
 - fill colors, changing, 97
 - for shape tween, 285
 - stroke color, changing, 98
 - text colors, changing, 169
- SWC files. *See* components

SWF files. *See* Flash movies
SWF History
 warning, 64-65
SWF Size History, 65, 500
switching between documents, 18
Symbol button, 46
symbols. *See also* buttons; instances;
 motion tweening; movie clips
 animations to symbol, converting, 139
 attributes, 463
 behavior, changing, 144
 duplicating during swap, 147
 editing
 entering/exiting editing mode,
 140-141
 master symbols, editing, 142-143
 existing artwork to symbol, converting,
 138
 exiting editing mode, 141
 Find and Replace for, 190
 font symbols, creating, 136-137
 graphic symbols, 148-149
 IK animations, 289-293
 master symbols
 breaking instances from, 155
 editing, 142-143
 new symbols creating, 138
 registration point, creating, 139
 warning preferences, setting, 65
syntax. *See also* dot syntax
 ActionScript 3.0, 377
 checking, 348
 indentation, 349
syntax-coloring
 in debugging, 413
 schemes, 349
system requirements, 2

T

tab size, setting, 349
telephones. *See also* cell phones
 activation requirement, 3
templates. *See also* documents
 for mobile device content, 512
 saving documents as, 23
10-second rule, 504

Test Movie, 217, 500. *See also* masks;
 motion tweening; shape tweening
 debugging with, 416-420
 general preferences, setting, 60
testing. *See also* debugging
 in AIR, 500
 debugging and, 411
 in Device Central, 500
 in Flash, 500
 frame-by-frame animation, 231
 preview, 499
 projects, 520
text, 153. *See also* characters; fonts; kerning
 aligning paragraph text, 172
 anti-aliasing, 182
 bitmap text, 182
 breaking text fields, 180-181
 columns, 174
 colors, changing, 169
 ComboBox component, adding, 434-435
 CSS (Cascading Style Sheets), working
 with, 191
 direction, 175
 dynamic text
 character options for, 182
 creating, 186
Illustrator, importing from, 196
input text, 163
 character options for, 182
 component, adding, 432-433
 creating, 186-187
 parameters, editing, 430-431
line spacing, setting, 173
margins, setting, 173
orientation, 176-177
Photoshop import preferences, 198-199
preferences, setting, 62
rotation of, 176-177
scrolling text, creating, 187
selecting, 172
shapes, breaking text fields into, 180-181
single column of vertical text, creating,
 175-176
spell-checking, 188-189
static text, 163
 creating, 164
TextArea component, adding, 440-441

- tracking, adjusting, 170
- transforming, 181
- vertical text, 175-177
 - block, creating, 177
 - direction, 175
 - options, setting, 62
 - orientation, 176-177
- text boxes, creating, 164
- text files, loading, 164
- Text tool, 164-165
- TextArea component, adding, 440-441
- TextInput component, adding, 432-433
- 3D Space
 - Global Transform, 122-123
 - perspective angle, 124
 - Rotation, tool, 122-123
 - vanishing point, 125
 - Translation, tool, 122
- thread and unthread text, TLF, 167
- TIFF file support, 195
- Timeline, 7, 30-31. *See also* effects; frames; sounds; video
 - changing display, 32
 - elements, adding motion to, 11
 - frames display, changing, 32-33
 - movie clips, referencing, 335
 - resizing display, 32
 - sync sounds to animation, adding, 309
- timing rule, 504
- tint color
 - instance tint color, modifying, 158
 - motion tween, adding to, 246
- TLF text
 - advanced options, 178-179
 - columns, creating, 174
 - creating, 165
 - direction, 175
 - linking text, 166-167
 - orientation, 176
- toggle buttons, adding, 442-443
- Toolbar, 6, 29, 30-31. *See also* Main toolbar
 - customizing, Tools panel, 50-51
 - expanding/collapsing, 49, 50
 - moving, 50
 - showing/hiding, 48-50, 52-53
- Toolbox, 348
- Tools, creating, 51

- Tools panel. *See* Toolbar
- tooltip preferences, 61
- trace actions, protecting from, 480
- Trace Bitmap dialog box, 216
- tracing bitmaps, 216
- tracking, adjusting, 170
- Transformation panel, 156-157
- transformation point, working with, 118
- transition settings, 490
- Translation tool, 3D, 122-123
- transparency
 - instance transparency, modifying, 159, 160
 - publishing movies, setting on, 489
- turning on/off
 - grids and guides, 91
 - Work Area visibility, 88
- tweening. *See* motion tweening; shape tweening
- typewriter fonts, 185

U

- UI components, connecting, 458
- uncompressed XFL file format, 20-21
- Undo command, 470
 - saving documents after using, 471
- undoing/redoing
 - general preferences, setting, 60
 - History panel for, 466, 470-471
 - number of levels, changing, 48
 - object level undo, working with, 468-469
 - scene deletions, 45
 - text transformations, 181
 - Transform effects, 158
- unlocking. *See* locking/unlocking
- Up state for buttons, 150
- updates
 - online updates/support, 26-27
 - previous Flash versions, 3
 - registering to receive, 26
- URLs changes warning, 64
- user interface components, 429

V

- variable names
 - for dynamic text, 186
 - for input text, 187

- vanishing point, 3D, 125
- variables, 364. *See also* ActionScripts
 - viewing, 421
 - Watch list, adding to, 422
- vector graphics, 73, 194
 - tracing bitmaps to creating, 216
- vectors, 73
- Version Detection, 487
- vertical lines, creating, 111
- vertical text. *See* text
- video, 299. *See also* Media Display; movie clips; QuickTime/QuickTime Pro
 - with behaviors, 334-335
 - components, 429
 - controlling from Timeline, 334-335
 - cue points (ActionScript), adding, 332-333
 - file formats for, 327
 - Find and Replace for, 190
 - FLV format, exporting clips into, 346
 - parameters, 328-329
 - Stage, moving file on, 328-330
 - Video Import wizard, using, 326-327
- video card requirements, 9
- Video Encoder, 340-341
- Video Import wizard, using, 326-327
- views and viewing
 - document properties, 16
 - History panel, changing view in, 473
 - Stage, changing view size of, 46
- Virtual Machine, 365

W

- warning preferences, setting, 64-65
- Watch list variables, adding, 422
- WAV files, exporting movies to, 502-503
- Web 216 color palette, 489

- Welcome screen, 4, 6
- While loops, 360
 - working with, 360
- width of gradient fills, changing, 121
- Windows Metafile (WMF)
 - support for, 195
- Windows PCs
 - keyboard commands on, 1
 - launching Flash, 4
 - printing documents, 72
 - projector, creating, 493
 - system requirements for, 2
- word wrapping, 412
- Work Area, 88
 - storing graphics in, 126
- Workspaces. *See* Panel Set.

X

- XFL
 - Opening, 14
 - Saving, 20-21
 - uncompressed file format, 20-21
- XML (Extensible Markup Language), 187
 - CSS (Cascading Style Sheets), 191
 - ECMAScript, 348
 - projects as, 518
 - saving, 20-21
- XMP file information, 484-485

Z

- Zoom button, 19
- zooming
 - on bitmap images, 202
 - changing magnification area, 87
 - with Magnifying Glass, 86-87
 - on sounds, 322