

Using ADOBE® FLASH® CS4 PROFESSIONAL



© 2008 Adobe Systems Incorporated. All rights reserved.

Using Adobe® Flash® CS4 Professional for Windows® and Mac OS

If this guide is distributed with software that includes an end user agreement, this guide, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by any such license, no part of this guide may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Adobe Systems Incorporated. Please note that the content in this guide is protected under copyright law even if it is not distributed with software that includes an end user license agreement.

The content of this guide is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.

Please remember that existing artwork or images that you may want to include in your project may be protected under copyright law. The unauthorized incorporation of such material into your new work could be a violation of the rights of the copyright owner. Please be sure to obtain any permission required from the copyright owner.

Any references to company names in sample templates are for demonstration purposes only and are not intended to refer to any actual organization.

Adobe, the Adobe logo, Adobe AIR, Adobe Premiere, ActionScript, After Effects, Authorware, ColdFusion, Creative Suite, Director, Dreamweaver, Fireworks, Flash, Flash Lite, Flex, Flex Builder, FrameMaker, FreeHand, Illustrator, InCopy, InDesign, Kuler, Pixel Bender, Photoshop, PostScript, Soundbooth, and Version Cue are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

ActiveX, Vista, and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Macintosh and Mac OS are trademarks of Apple Inc. registered in the U.S. and other countries. Java is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries. Intel is a registered trademark of Intel Corporation in the U.S. and other countries. Helvetica is a trademark of Heidelberger Druckmaschinen AG exclusively licensed through Linotype Library GmbH, and may be registered in certain jurisdictions. Arial is a trademark of The Monotype Corporation registered in the U.S. Patent and Trademark Office and certain other jurisdictions. All other trademarks are the property of their respective owners.

This product includes software developed by the Apache Software Foundation (www.apache.org).

MPEG Layer-3 audio compression technology licensed by Fraunhofer IIS and Thomson Multimedia (<http://www.iis.fhg.de/amm/>). You cannot use the mp3 compressed audio within the Software for real time or live broadcasts. If you require an mp3 decoder for real time or live broadcasts, you are responsible for obtaining this mp3 technology license.

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

Video compression and decompression is powered by On2 TrueMotion video technology. © 1992-2005 On2 Technologies, Inc. All Rights Reserved. <http://www.on2.com>.

This product includes software developed by the OpenSymphony Group (<http://www.opensymphony.com/>)

This product contains either BSAFE and/or TIPEM software by RSA Security, Inc.



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

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA.

Notice to U.S. Government End Users: The Software and Documentation are "Commercial Items," as that term is defined at 48 C.F.R. §2.101, consisting of "Commercial Computer Software" and "Commercial Computer Software Documentation," as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §§227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Unpublished-rights reserved under the copyright laws of the United States. Adobe agrees to comply with all applicable equal opportunity laws including, if appropriate, the provisions of Executive Order 11246, as amended, Section 402 of the Vietnam Era Veterans Readjustment Assistance Act of 1974 (38 USC 4212), and Section 503 of the Rehabilitation Act of 1973, as amended, and the regulations at 41 CFR Parts 60-1 through 60-60, 60-250, and 60-741. The affirmative action clause and regulations contained in the preceding sentence shall be incorporated by reference.

Contents

Chapter 1: Resources

Activation and registration	1
Help and support	2
Services, downloads, and extras	3
Getting started in Flash	4
What's new	6

Chapter 2: Workspace

Flash workflow and workspace	9
Working with ConnectNow	16
Using the Stage and Tools panel	17
The Timeline	22
Using Flash authoring panels	25
Undo, redo, and history	29
Automating tasks with the Commands menu	32
Accessibility in the Flash workspace	33
Set preferences in Flash	37
Keyboard shortcuts	40

Chapter 3: Managing documents

Working with Flash documents	43
Printing Flash documents	50
Working with projects	52
Find and Replace	58
Templates	62

Chapter 4: Using imported artwork

Placing artwork into Flash	63
Working with Fireworks files	67
Working with Illustrator AI files	69
Working with Photoshop PSD files	78
Imported bitmaps	86

Chapter 5: Creating and Editing Artwork

Drawing	91
Reshaping objects	111
Moving, arranging, and deleting artwork	118
Colors, gradients, and strokes	126
3D graphics	139

Chapter 6: Symbols, instances, and library assets

Working with symbols	147
Working with symbol instances	152
Working with the library	157

Sharing library assets	162
Working with button symbols	164
Scaling and caching symbols	166
Symbols and ActionScript	168
Chapter 7: Timelines and Animation	
Working with timelines	172
Working with scenes	178
Timelines and ActionScript	180
Animation basics	183
Motion tweens	187
Working with classic tweens	210
Shape tweening	218
Frame-by-frame animation	220
Using mask layers	221
Using inverse kinematics	223
Chapter 8: Filters and Blends	
About filters	234
Applying filters	235
Applying blend modes	242
Chapter 9: Text	
About text and fonts	244
Creating text	246
Setting text attributes	254
About multilanguage text	258
Creating multilanguage text	263
XML file format for multilanguage text	267
Multilanguage text and ActionScript	270
Chapter 10: Sound	
Using sounds in Flash	273
Exporting Sounds	278
Sound and ActionScript	280
Chapter 11: Video	
Digital video and Flash	283
Import video for streaming or progressive download	288
Embed video in a SWF file	290
Export QuickTime video files	293
Working with Adobe Premiere Pro and After Effects	294
Controlling external video playback with ActionScript	296
Chapter 12: Creating e-learning content	
Getting started with Flash e-learning	301

Chapter 13: Creating accessible content	
About accessible content	302
Using Flash to enter accessibility information for screen readers	305
Specifying advanced accessibility options for screen readers	310
Creating accessibility with ActionScript	312
Chapter 14: Working with screens	
Screen-based documents and the screen authoring environment	316
Working with screens	318
Adding content to screens	323
Chapter 15: ActionScript	
Working with ActionScript	330
Script Assist mode and behaviors	337
Writing and managing scripts	339
Debugging ActionScript 1.0 and 2.0	350
Debugging ActionScript 3.0	360
Working with Flash and Flex	364
Printing at runtime	366
ActionScript publish settings	371
Chapter 16: Publishing and Exporting	
Publishing Flash documents	379
Publish settings	383
Publishing for Adobe AIR	398
HTML publishing templates	406
About exporting from Flash	410
Exporting Images and Graphics	412
Exporting video and sound	415
Chapter 17: Best practices	
Structuring FLA files	418
Organizing ActionScript in an application	420
Behaviors conventions	421
Video conventions	423
SWF application authoring guidelines	425
Accessibility guidelines	429
Advertising with Flash	434
Optimizing FLA files for SWF output	436
Tips for creating content for mobile devices	445
Index	453

Chapter 1: Resources

Before you begin working with your software, take a few moments to read an overview of activation and the many resources available to you. You have access to instructional videos, plug-ins, templates, user communities, seminars, tutorials, RSS feeds, and much more.

Activation and registration

Help with installation

For help with installation issues, see the Installation Support Center at www.adobe.com/go/cs4install.

License activation

During the installation process, your Adobe software contacts Adobe to complete the license activation process. No personal data is transmitted. For more information on product activation, visit the Adobe website at www.adobe.com/go/activation.

A single-user retail license activation supports two computers. For example, you can install the product on a desktop computer at work and on a laptop computer at home. If you want to install the software on a third computer, first deactivate it on one of the other two computers. Choose Help > Deactivate.

Register

Register your product to receive complimentary installation support, notifications of updates, and other services.

- ❖ To register, follow the on-screen instructions in the Registration dialog box, which appears after you install the software.



If you postpone registration, you can register at any time by choosing Help > Registration.

Adobe Product Improvement Program

After you use your Adobe software a certain number of times, a dialog box may appear asking whether you want to participate in the Adobe Product Improvement Program.

If you choose to participate, data about your use of Adobe software is sent to Adobe. No personal information is recorded or sent. The Adobe Product Improvement Program only collects information about which features and tools you use and how often you use them.

You can opt in to or out of the program at any time:

- To participate, choose Help > Adobe Product Improvement Program and click Yes, Participate.
- To stop participating, choose Help > Adobe Product Improvement Program and click No, Thank You.

ReadMe

A ReadMe file for your software is available online and on the installation disc. Open the file to read important information about topics such as the following:

- System requirements
- Installation (including uninstalling the software)
- Activation and registration
- Font installation
- Troubleshooting
- Customer support
- Legal notices

Help and support

Community Help

Community Help is an integrated environment on Adobe.com that gives you access to community-generated content moderated by Adobe and industry experts. Comments from users help guide you to an answer.

Community Help draws on a number of resources, including:

- Videos, tutorials, tips and techniques, blogs, articles, and examples for designers and developers.
- Complete online product Help, which is updated regularly by the Adobe documentation team.
- All other content on Adobe.com, including knowledgebase articles, downloads and updates, Developer Connection, and more.

Choose Help > *product name* Help in the application to access the Help and Support page, the portal to all of the Community Help content for your product. You can also use the Help search field in some Creative Suite 4 applications, or press F1 (Windows), to access Community Help for your product.

The sites searched by the default Community Help search engine are hand-selected and reviewed for quality by Adobe and Adobe Community Experts. Adobe experts also work to ensure that the top search results include a mixture of different kinds of content, including results from online product Help.

For more information on using Community Help, see http://help.adobe.com/en_US/CommunityHelp/.

For a video overview of Community Help, see www.adobe.com/go/lrvid4117_xp.

Product Help

Adobe provides a comprehensive user guide for each product in several formats, including online product Help, PDF, and printed book. Results from online product Help are included in your results whenever you search Community Help.


If you're connected to the Internet, the Help menu within the product opens the product Help and Support page by default. This page is a portal to all of the Community Help content for the product. If you want to consult or search online product Help only, you can access it by clicking the product Help link in the upper-right corner of the Help and Support page. Be sure to select the **This Help System Only** option before you do your search.

If you're not connected to the Internet, the Help menu within the product opens local Help, a subset of the content available in online product Help. Because local Help is not as complete or up-to-date as online product Help, Adobe recommends that you use the PDF version of product Help if you want to stay offline. A downloadable PDF of complete product Help is available from two places:

- The product's Help and Support page (upper-right corner of the page)
- Local and web Help (top of the Help interface)

For more information on accessing product help, see http://help.adobe.com/en_US/CommunityHelp/.

If you are working in Adobe InDesign, Photoshop, Illustrator, Flash, Fireworks, or Dreamweaver, and you want to turn off Community Help so that local Help opens by default, do the following:

- 1 Open the Connections panel (Window > Extensions > Connections).
- 2 From the Connections panel menu , select Offline Options.
- 3 Select Keep Me Offline and click OK.

Note: When you disable web services from the Connections panel, all other web services (such as Adobe Kuler and Adobe ConnectNow) are also disabled.

Printed resources

Printed versions of the complete online product Help are available for the cost of shipping and handling at www.adobe.com/go/store.

Support resources

Visit the Adobe Support website at www.adobe.com/support to learn about free and paid technical support options.

Services, downloads, and extras

You can enhance your product by integrating a variety of services, plug-ins, and extensions in your product. You can also download samples and other assets to help you get your work done.

Adobe creative online services

Adobe® Creative Suite® 4 includes new online features that bring the power of the web to your desktop. Use these features to connect with the community, collaborate, and get more from your Adobe tools. Powerful creative online services let you complete tasks ranging from color matching to data conferencing. The services seamlessly integrate with desktop applications so you can quickly enhance existing workflows. Some services offer full or partial functionality when you're offline too.

Visit Adobe.com to learn more about available services. Some Creative Suite 4 applications include these initial offerings:

Kuler™ panel Quickly create, share, and explore color themes online.

Adobe® ConnectNow Collaborate with dispersed working teams over the web, sharing voice, data, and multimedia.

Resource Central Instantly access tutorials, sample files, and extensions for Adobe digital video applications.

For information on managing your services, see the Adobe website at www.adobe.com/go/learn_creativeservices_en.

Adobe Exchange

Visit the Adobe Exchange at www.adobe.com/go/exchange to download samples as well as thousands of plug-ins and extensions from Adobe and third-party developers. The plug-ins and extensions can help you automate tasks, customize workflows, create specialized professional effects, and more.

Adobe downloads

Visit www.adobe.com/go/downloads to find free updates, tryouts, and other useful software.

Adobe Labs

Adobe Labs at www.adobe.com/go/labs gives you the opportunity to experience and evaluate new and emerging technologies and products from Adobe. At Adobe Labs, you have access to resources such as these:

- Prerelease software and technologies
- Code samples and best practices to accelerate your learning
- Early versions of product and technical documentation
- Forums, wiki-based content, and other collaborative resources to help you interact with like-minded users.

Adobe Labs fosters a collaborative software development process. In this environment, customers quickly become productive with new products and technologies. Adobe Labs is also a forum for early feedback. The Adobe development teams use this feedback to create software that meets the needs and expectations of the community.

Adobe TV

Visit Adobe TV at <http://tv.adobe.com> to view instructional and inspirational videos.

Extras

The installation disc contains a variety of extras to help you make the most of your Adobe software. Some extras are installed on your computer during the setup process; others are located on the disc.

To view the extras installed during the setup process, navigate to the application folder on your computer.

- Windows®: *[startup drive]\Program Files\Adobe\[Adobe application]*
- Mac OS®: *[startup drive]/Applications/[Adobe application]*

To view the extras on the disc, navigate to the Goodies folder in your language folder on the disc. Example:

- */English/Goodies/*

Getting started in Flash

If you're new to Flash, get a good overview by viewing these videos.

Learn Flash Professional CS4 - Getting Started Series

[Understanding Flash](#)

[Understanding Flash file types](#)

[Exploring the Flash interface](#)

[Setting up workspaces](#)
[Using the drawing tools](#)
[Object and merge drawing](#)
[Understanding symbols](#)
[Working with bitmap graphics](#)
[Working with sound](#)
[Working with video](#)
[Understanding tweens](#)
[Creating shape tweens](#)
[Creating motion tweens](#)
[Using the Motion Editor](#)
[Working with motion presets](#)
[Animating bones \(inverse kinematics\)](#)
[ActionScript 3.0](#)
[Building an application](#)
[Creating a Flash web movie](#)

Learn Flash Professional CS4 Series

[Introducing Flash CS4 Professional](#)
[Managing Design Suite Workspaces \(Demo is in Photoshop\)](#)
[Using Motion Presets](#)
[Creating motion tweens: Part 1](#)
[Creating motion tweens: Part 2](#)
[Using the Motion Editor](#)
[Using inverse kinematics](#)
[Working with 3D art](#)
[Using the decorative drawing tools](#)
[Understanding Flash Integration \(InDesign import\)](#)
[Using InDesign Export to Flash \(XFL\)](#)
[Exporting artboards to Flash \(Illustrator import\)](#)
[Working with Soundbooth and Flash](#)
[Exporting from After Effects to Flash via XFL](#)
[Using Kuler color themes](#)
[Using ConnectNow to share your screen \(Demo is in Dreamweaver\)](#)
[Introducing Adobe Device Central CS4](#)

What's new

New features

The following features are new to Adobe® Flash® CS4 Professional.

Object-based animation

Object-based animation dramatically simplifies designing in Flash while simultaneously providing greater control. Tweens are now applied directly to objects instead of keyframes, providing precise control over each of the individual animation attributes. For more information, see [“Motion tweens”](#) on page 187.

Motion Editor panel

Experience complete independent control over every keyframe parameter including rotation, size, scale, position, filters, and more using the keyframe editor. Use the keyframe editor to graphically control easing using curves. For more information, see [“Editing property curves with the Motion Editor”](#) on page 203.

Motion Tween Presets

Apply pre-built animations to any object to jumpstart your project. Select from dozens of pre-built presets, or create and save your own. Share presets across teams to save animation time. For more information, see [“Applying motion presets”](#) on page 189.

Inverse Kinematics with the Bone tool

Easily create chain-like effects with a series of linked objects or quickly distort single objects using the Bones tool. For more information, see [“Using inverse kinematics”](#) on page 223.

3D Transformation

Animate 2D objects through 3D space with new 3D transformation tools. Transformation tools including rotate and translate allow you to animate across X, Y, and Z axes. Apply local or global rotation to rotate an object within relationship to itself or to the Stage. For more information, see [“3D graphics”](#) on page 139.

Decorative drawing with the Deco tool

Easily turn any symbol into an instant design tool. Whether creating patterns that can then be applied using the brush or fill tools, or creating kaleidoscope-like effects using single or multiple symbols with the deco symmetry tool, Deco provides a new way to design with symbols. For more information, see [“Drawing patterns with the Decorative drawing tools”](#) on page 108.

Adobe Kuler panel

The Kuler panel is your portal to groups of colors, or themes, created by an online community of designers. You can use it to browse thousands of themes on the Kuler web site, and then download selected themes to edit or to include in your own projects. You can also use the Kuler panel to create and save themes, and then share them with the Kuler community. For more information, see [“Kuler panel”](#) on page 137.

Authoring for Adobe AIR

Deliver interactive experiences to the desktop with new Publish to AIR. Adobe® AIR™, a new cross-operating system runtime allows you to leverage local desktop resources and data to deliver even more personal, engaging experiences. Using the same skills you use to deliver to the Flash® Player, your content can now reach even more audiences across more devices – web, mobile and now the desktop. For more information, see [“Publishing for Adobe AIR”](#) on page 398.

Sample Sounds library

A new included library of sound effects makes creating content with sound even easier. For more information, see [“Importing sounds”](#) on page 273.

Vertical Property inspector

The Property inspector is now vertically displayed taking better advantage of wider screens to provide you with more Stage real estate.

New Project Panel

Work with multiple-file projects more easily with the new project panel. Apply property changes to multiple files, save symbols to a designated file folder at the time they are created, and more. For more information, see [“Working with projects”](#) on page 52.

Adobe Media Encoder with H.264 support

Deliver the highest quality video with more control than ever before. Encode to any format recognized by Adobe Flash Player with the same tool found in other Adobe video products such as Adobe® Premiere® Pro and After Effects®. For more information, see Adobe Media Encoder help.

Adobe ConnectNow integration

Adobe® ConnectNow™ lets you share your screen and meet with other users online. You can open ConnectNow directly from the application interface by choosing File > Share My Screen. For more information, see [Working with ConnectNow](#).

Edit in Soundbooth

Users of Adobe® Soundbooth™ can now edit imported sounds in Soundbooth directly from within Flash. The Adobe ASND sound file format enables non-destructive editing of sound files. For more information, see [“Edit a sound in Soundbooth”](#) on page 277.

Enhanced metadata support

New XMP panel enables users to quickly and easily assign metadata tags to their SWF content. Enables addition of metadata into SWF files that is recognized by Adobe® Bridge and other Creative Suite® applications that recognize XMP metadata. Improves organization and enables fast find and retrieval of SWF files. For more information, see [“Add XMP metadata to a document”](#) on page 45.

Collaborate with Flex developers

Import code-only ActionScript® 3.0 SWC components developed in Flex™ Builder™. For more information, see [“Creating components for Flex”](#) on page 364.

XFL import

Import content from InDesign® and After Effects with complete file integrity. XFL is a new file format which enables designers to easily deliver content for further development in Flash. For more information see [“Opening XFL files”](#) on page 47.

Support for Adobe Pixel Bender

Easily apply custom filters and effects at runtime for real-time creative expression. Use Adobe® Pixel Bender™ to create unique filters, blends, and fills - which are now portable to multiple applications with Adobe Pixel Bender - to produce compelling, on-demand visual effects for the Adobe Flash Player runtime environment. Write a filter once, and then share it or apply it multiple times using ActionScript 3.0.

JPEG deblocking

This publish settings option reduces the appearance of the common artifacts found in highly compressed JPEG files. For more information, see [“Specify publish settings for SWF files”](#) on page 383.

Improved Library panel

Work with assets more easily, thanks to a new, improved Library panel that includes search capability, sorting, and the ability to set properties on multiple library items at once. For more information, see [“About the Library panel”](#) on page 26.

New Creative Suite user interface

Boost your efficiency with intuitive panel docking and spring-loaded behaviors that help streamline your interaction with tools throughout Adobe’s Creative Suites. For more information, see [“Workspace”](#) on page 9.

New Font menus

The font menu in Flash now includes previews of each font and each style included in the font. For more information, see [“Setting text attributes”](#) on page 254.

Hardware Acceleration

Published SWF files can now take better advantage of available hardware for better playback performance. For more information, see [“Specify publish settings for SWF files”](#) on page 383.

Community Help

Community Help is an integrated environment on adobe.com that gives you access to community-generated content moderated by Adobe and industry experts. Comments and ratings from users help guide you to an answer. Search Community Help to find the best content on the web about Adobe products and technologies. For more information, see [“Help and support”](#) on page 2.

Chapter 2: Workspace

The Adobe® Flash® CS4 Professional workspace includes tools and panels that help you create and navigate your documents. Understanding these tools will help you maximize the application's capabilities.

Flash workflow and workspace

General Flash workflow

To build a Flash application, you typically perform the following basic steps:

Plan the application.

Decide which basic tasks the application will perform.

Add media elements.

Create and import media elements, such as images, video, sound, and text.

Arrange the elements.

Arrange the media elements on the Stage and in the Timeline to define when and how they appear in your application.

Apply special effects.

Apply graphic filters (such as blurs, glows, and bevels), blends, and other special effects as you see fit.

Use ActionScript to control behavior.

Write ActionScript® code to control how the media elements behave, including how the elements respond to user interactions.

Test and publish your application.

Test to verify that your application is working as you intended and find and fix any bugs you encounter. You should test the application throughout the creation process. Publish your FLA file as a SWF file that can be displayed in a web page and played back with Flash® Player.

Depending on your project and your working style, you might use these steps in a different order.

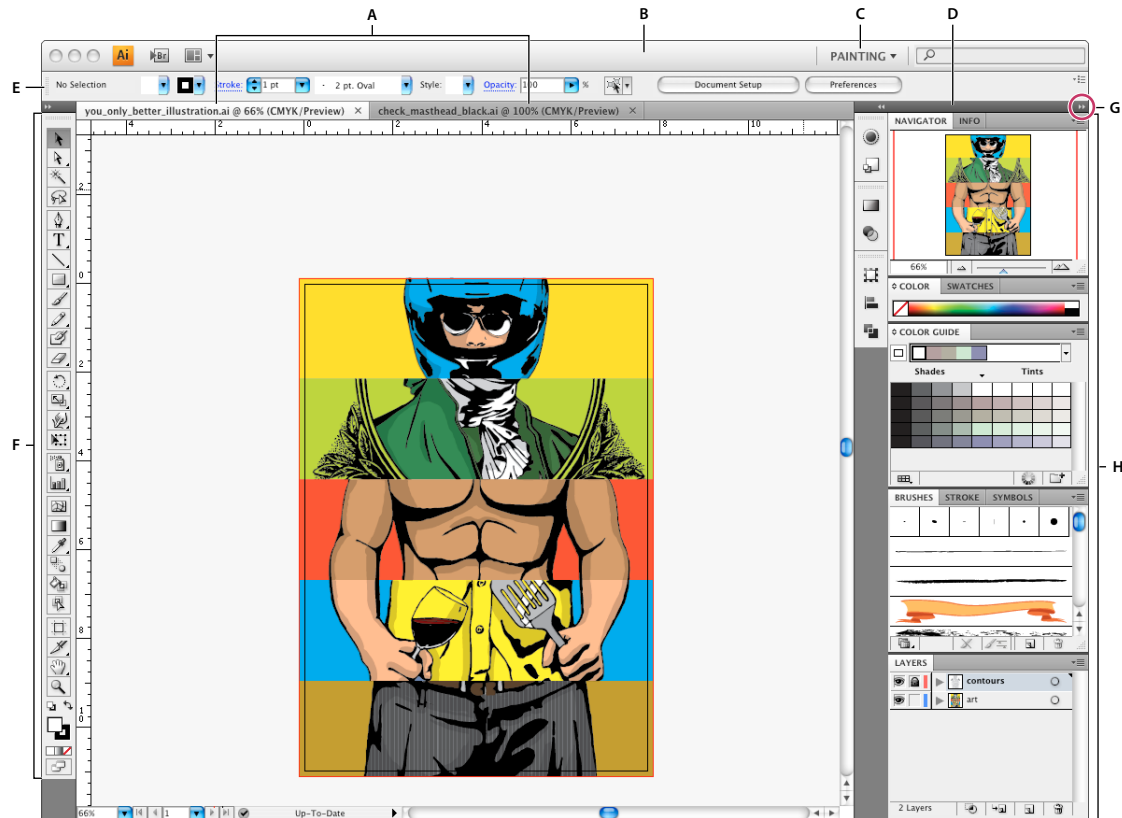
For more help getting started with the Flash workflow, see the following:

- Article: *Introducing Adobe Flash CS4 Professional*: http://www.adobe.com/devnet/logged_in/dwinnie_fics4.html
- Article: *Creating a simple document in Flash CS4 Professional*: http://www.adobe.com/devnet/flash/articles/flash_cs4_createfla.html
- Video: *Flash CS4 Workflow Basics*: http://www.adobe.com/go/lrvid4053_fl

Workspace overview

You create and manipulate your documents and files using various elements, such as panels, bars, and windows. Any arrangement of these elements is called a *workspace*. The workspaces of the different applications in Adobe® Creative Suite® 4 share the same appearance so that you can move between the applications easily. You can also adapt each application to the way you work by selecting from several preset workspaces or by creating one of your own.

Although the default workspace layout varies in different products, you manipulate the elements much the same way in all of them.



Default Illustrator workspace


A. Tabbed Document windows B. Application bar C. Workspace switcher D. Panel title bar E. Control panel F. Tools panel G. Collapse To Icons button H. Four panel groups in vertical dock

- The *Application bar* across the top contains a workspace switcher, menus (Windows only), and other application controls. On the Mac for certain products, you can show or hide it using the Window menu.
- The *Tools panel* contains tools for creating and editing images, artwork, page elements, and so on. Related tools are grouped.
- The *Control panel* displays options for the currently selected tool. The Control panel is also known as the options bar in Photoshop. (Adobe Flash®, Adobe Dreamweaver®, and Adobe Fireworks® have no Control panel.)
- Flash, Dreamweaver, and Fireworks have a *Property inspector* that displays options for the currently selected element or tool.
- The *Document window* displays the file you're working on. Document windows can be tabbed and, in certain cases, grouped and docked.

- *Panels* help you monitor and modify your work. Examples include the Timeline in Flash, the Layers panel in Adobe Photoshop®, and the CSS Styles panel in Dreamweaver. Panels can be grouped, stacked, or docked.
- On the Mac, the *Application frame* groups all the workspace elements in a single, integrated window that lets you treat the application as a single unit. When you move or resize the Application frame or any of its elements, all the elements within it respond to each other so none overlap. Panels don't disappear when you switch applications or when you accidentally click out of the application. If you work with two or more applications, you can position each application side by side on the screen or on multiple monitors. If you prefer the traditional, free-form user interface of the Mac, you can turn off the Application frame. In Adobe Illustrator®, for example, select Window > Application Frame to toggle it on or off. (In Flash, the Application frame is on permanently. Dreamweaver does not use an Application frame.)


Hide or show all panels


- (Illustrator, Adobe InCopy®, Adobe InDesign®, Photoshop, Fireworks) To hide or show all panels, including the Tools panel and Control panel, press Tab.
- (Illustrator, InCopy, InDesign, Photoshop) To hide or show all panels except the Tools panel and Control panel, press Shift+Tab.

 You can temporarily display hidden panels if Auto-Show Hidden Panels is selected in Interface preferences. It's always on in Illustrator. Move the pointer to the edge of the application window (Windows®) or to the edge of the monitor (Mac OS®) and hover over the strip that appears.

- (Flash, Dreamweaver, Fireworks) To hide or show all panels, press F4.

Display panel options

- ❖ Click the panel menu icon  in the upper-right corner of the panel.


 You can open a panel menu even when the panel is minimized.

(Illustrator) Adjust panel brightness

- ❖ In User Interface preferences, move the Brightness slider. This control affects all panels, including the Control panel.

Reconfigure the Tools panel

You can display the tools in the Tools panel in a single column, or side by side in two columns. (This feature is not available in the Tools panel in Fireworks and Flash.)

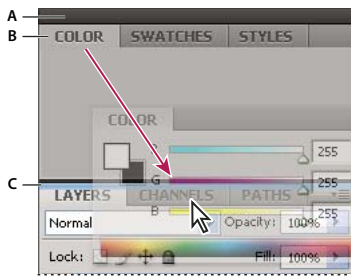
 In InDesign and InCopy, you also can switch from single-column to double-column (or single-row) display by setting an option in Interface preferences.

- ❖ Click the double arrow at the top of the Tools panel.


Manage windows and panels

You can create a custom workspace by moving and manipulating Document windows and panels. You can also save workspaces and switch among them.

Note: The following examples use Photoshop for demonstration purposes. The workspace behaves the same in all the products.



Narrow blue drop zone indicates Color panel will be docked on its own above the Layers panel group.
 A. Title bar B. Tab C. Drop zone

 In Photoshop, you can change the font size of the text in the Control panel, in the panels, and in tool tips. Choose a size from the UI Font Size menu in Interface preferences.

Manage Document windows

When you open more than one file, the Document windows are tabbed.

- To rearrange the order of tabbed Document windows, drag a window's tab to a new location in the group.
- To undock a Document window from a group of windows, drag the window's tab out of the group.
- To dock a Document window to a separate group of Document windows, drag the window into the group.

Note: Dreamweaver does not support docking and undocking Document windows. Use the Document window's Minimize button to create floating windows.

- To create groups of stacked or tiled documents, drag the window to one of the drop zones along the top, bottom, or sides of another window. You can also select a layout for the group by using the Layout button on the Application bar.

Note: Some products do not support this functionality. However, your product may have Cascade and Tile commands in the Window menu to help you lay out your documents.

- To switch to another document in a tabbed group when dragging a selection, drag the selection over the document's tab for a moment.

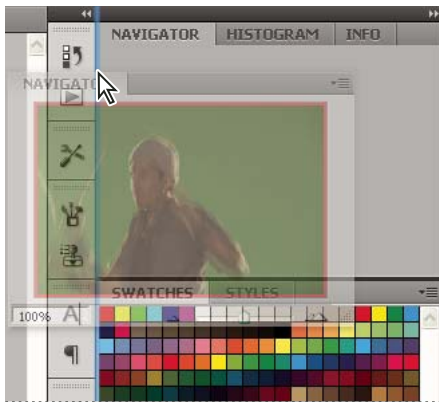
Note: Some products do not support this functionality.

Dock and undock panels

A *dock* is a collection of panels or panel groups displayed together, generally in a vertical orientation. You dock and undock panels by moving them into and out of a dock.

Note: Docking is not the same as stacking. A *stack* is a collection of floating panels or panel groups, joined top to bottom.

- To dock a panel, drag it by its tab into the dock, at the top, bottom, or in between other panels.
- To dock a panel group, drag it by its title bar (the solid empty bar above the tabs) into the dock.
- To remove a panel or panel group, drag it out of the dock by its tab or title bar. You can drag it into another dock or make it free-floating.



Navigator panel being dragged out to new dock, indicated by blue vertical highlight



Navigator panel now in its own dock

💡 You can prevent panels from filling all the space in a dock. Drag the bottom edge of the dock up so it no longer meets the edge of the workspace.

Move panels

As you move panels, you see blue highlighted *drop zones*, areas where you can move the panel. For example, you can move a panel up or down in a dock by dragging it to the narrow blue drop zone above or below another panel. If you drag to an area that is not a drop zone, the panel floats freely in the workspace.

- To move a panel, drag it by its tab.
- To move a panel group or a stack of floating panels, drag the title bar.

💡 Press **Ctrl** (Windows) or **Command** (Mac OS) while moving a panel to prevent it from docking. Press **Esc** while moving the panel to cancel the operation.

Note: The dock is stationary and can't be moved. However, you can create panel groups or stacks and move them anywhere.

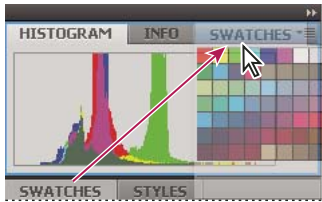
Add and remove panels

If you remove all panels from a dock, the dock disappears. You can create a dock by moving panels to the right edge of the workspace until a drop zone appears.

- To remove a panel, right-click (Windows) or Control-click (Mac) its tab and then select Close, or deselect it from the Window menu.
- To add a panel, select it from the Window menu and dock it wherever you want.

Manipulate panel groups

- To move a panel into a group, drag the panel's tab to the highlighted drop zone in the group.

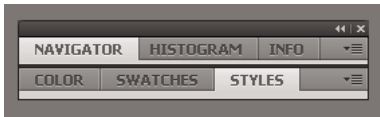


Adding a panel to a panel group

- To rearrange panels in a group, drag a panel's tab to a new location in the group.
- To remove a panel from a group so that it floats freely, drag the panel by its tab outside the group.
- To move a group, drag the title bar (the area above the tabs).

Stack floating panels

When you drag a panel out of its dock but not into a drop zone, the panel floats freely. The floating panel allows you to position it anywhere in the workspace. You can stack floating panels or panel groups so that they move as a unit when you drag the topmost title bar. (Panels that are part of a dock cannot be stacked or moved as a unit in this way.)



Free-floating stacked panels

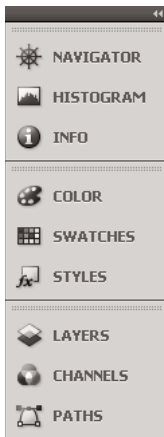
- To stack floating panels, drag a panel by its tab to the drop zone at the bottom of another panel.
 - To change the stacking order, drag a panel up or down by its tab.
- Note:** Be sure to release the tab over the narrow drop zone between panels, rather than the broad drop zone in a title bar.
- To remove a panel or panel group from the stack, so that it floats by itself, drag it out by its tab or title bar.

Resize panels

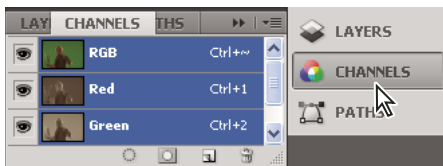
- To minimize or maximize a panel, panel group, or stack of panels, double-click a tab. You can also single-click the tab area (the empty space next to the tabs).
- To resize a panel, drag any side of the panel. Some panels, such as the Color panel in Photoshop, cannot be resized by dragging.

Manipulate panels collapsed to icons

You can collapse panels to icons to reduce clutter on the workspace. In some cases, panels are collapsed to icons in the default workspace.




Panels collapsed to icons



Panels expanded from icons

- To collapse or expand all panel icons in a dock, click the double arrow at the top of the dock.
- To expand a single panel icon, click it.
- To resize panel icons so that you see only the icons (and not the labels), adjust the width of the dock until the text disappears. To display the icon text again, make the dock wider.
- To collapse an expanded panel back to its icon, click its tab, its icon, or the double arrow in the panel's title bar.

 *In some products, if you select Auto-Collapse Icon Panels from the Interface or User Interface Options preferences, an expanded panel icon collapses automatically when you click away from it.*

- To add a floating panel or panel group to an icon dock, drag it in by its tab or title bar. (Panels are automatically collapsed to icons when added to an icon dock.)
- To move a panel icon (or panel icon group), drag the icon. You can drag panel icons up and down in the dock, into other docks (where they appear in the panel style of that dock), or outside the dock (where they appear as floating, expanded panels).

Restore the default workspace

- Select the default, Essentials workspace from the workspace switcher in the Application bar.
- (Photoshop) Select Window > Workspace > Essentials (Default).
- (InDesign, InCopy) Select Window > Workspace > Reset [*Workspace Name*].

Save and switch workspaces

By saving the current size and position of panels as a named workspace, you can restore that workspace even if you move or close a panel. The names of saved workspaces appear in the workspace switcher in the Application bar.

In Photoshop, the saved workspace can include a specific keyboard shortcut set and menu set.

Save a custom workspace

- 1 With the workspace in the configuration you want to save, do one of the following:
 - (Photoshop, Illustrator) Choose Window > Workspace > Save Workspace.
 - (InDesign, InCopy) Choose Window > Workspace > New Workspace.
 - (Dreamweaver) Choose Window > Workspace Layout > New Workspace.
 - (Flash) Choose New Workspace from the workspace switcher in the Application bar.
 - (Fireworks) Choose Save Current from the workspace switcher in the Application bar.
- 2 Type a name for the workspace.
- 3 (Photoshop, InDesign) Under Capture, select one or more options:
 - Panel Locations** Saves the current panel locations.
 - Keyboard shortcuts** Saves the current set of keyboard shortcuts (Photoshop only).
 - Menus** Saves the current set of menus.
- 4 Click OK or Save.

Display or switch workspaces

- ❖ Select a workspace from the workspace switcher in the Application bar.



In Photoshop, you can assign keyboard shortcuts to each workspace to navigate among them quickly.

Delete a custom workspace

- Select Manage Workspaces from the workspace switcher in the Application bar, select the workspace, and then click Delete. (The option is not available in Fireworks.)
- (Photoshop, InDesign, InCopy) Select Delete Workspace from the workspace switcher.
- (Illustrator) Choose Window > Workspace > Manage Workspaces, select the workspace, and then click the Delete icon.
- (InDesign) Choose Window > Workspace > Delete Workspace, select the workspace, and then click Delete.

(Photoshop) Start with the last or default panel locations

When you start Photoshop, panels can either appear in their original default locations, or appear as you last used them.

In Interface preferences:

- To display panels in their last locations on startup, select Remember Panel Locations.
- To display panels in their default locations on startup, deselect Remember Panel Locations.

Working with ConnectNow

Adobe® ConnectNow™ lets you share your screen and meet with other users online. You can open ConnectNow directly from the application interface by choosing File > Share My Screen.

Working with ConnectNow

Adobe® ConnectNow provides you with a secure, personal online meeting room where you can meet and collaborate with others via the web in real time. With ConnectNow, you can share and annotate your computer screen, send chat messages, and communicate using integrated audio. You can also broadcast live video, share files, capture meeting notes, and control an attendee's computer.

You can access ConnectNow directly from the application interface.

- 1 Choose File > Share My Screen.
- 2 In the Share My Screen dialog box, enter your Adobe ID and password, and click Sign In. If you don't have an Adobe ID and password, click the Create a Free Adobe ID link at the top of the dialog box.
- 3 To share your screen, click the Share My Computer Screen button at the center of the ConnectNow application window.

For complete instructions on using ConnectNow, see http://help.adobe.com/en_US/Acrobat.com/ConnectNow/index.html.

Using the Stage and Tools panel

Welcome screen overview

When Flash is running with no documents open, the Welcome screen appears. The Welcome screen contains the following four areas:

Open a Recent Item Lets you open your most recent documents (click the Open icon).

Create New Lists Flash file types, such as Flash documents and ActionScript® files.

Create from Template Lists the templates most commonly used to create Flash documents.

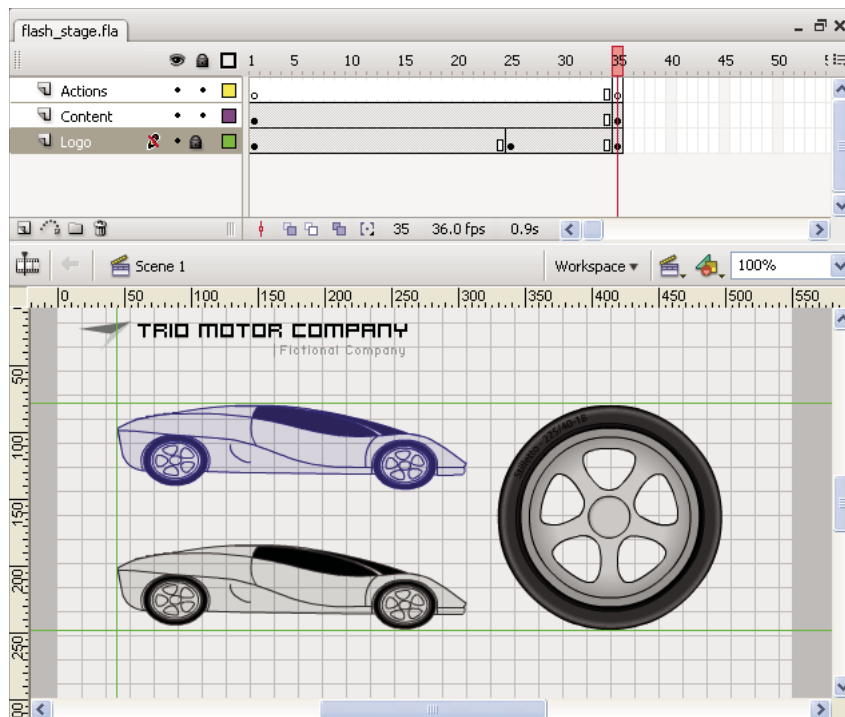
Extend Links to the Flash Exchange website, where you can download helper applications, extensions, and related information.

The Welcome screen also offers quick access to Help resources. You can take a tour of Flash, learn about documentation resources, and find Adobe Authorized Training facilities.

- To hide the Welcome screen, select Don't Show Again.
- To show the Welcome screen, select Edit > Preferences (Windows®) or select Flash > Preferences (Macintosh®), and select Welcome Screen from the On Launch menu in the General category.

Using the Stage




The Stage is the rectangular area where you place graphic content when creating Flash documents. The Stage in the authoring environment represents the rectangular space in Flash Player or in a web browser window where your document appears during playback. To change the view of the Stage as you work, zoom in and out. To help you position items on the Stage, you can use the grid, guides, and rulers.



The Timeline and Stage with content.

Zoom the Stage

To view the entire Stage on the screen, or to view a particular area of your drawing at high magnification, change the magnification level. The maximum magnification depends on the resolution of your monitor and the document size. The minimum value for zooming out on the Stage is 8%. The maximum value for zooming in on the Stage is 2000%.

- To zoom in on an element, select the Zoom tool  in the Tools panel, and click the element. To switch the Zoom tool between zooming in or out, use the Enlarge  or Reduce  modifiers (in the options area of the Tools panel when the Zoom tool is selected) or Alt-click (Windows) or Option-click (Macintosh).
- To zoom in so that a specific area of your drawing fills the window, drag a rectangular selection on the Stage with the Zoom tool.
- To zoom in on or out of the entire Stage, select View > Zoom In or View > Zoom Out.
- To zoom in or out by a specified percentage, select View > Magnification, and select a percentage from the submenu or select a percentage from the Zoom control at the upper-right corner of the document window.
- To scale the Stage so that it fits completely in the application window, select View > Magnification > Fit in Window.
- To show the contents of the current frame, select View > Magnification > Show All, or select Show All from the Zoom control at the upper-right side of the application window. If the scene is empty, the entire Stage appears.
- To show the entire Stage, select View > Magnification > Show Frame or select Show Frame from the Zoom control at the upper-right corner of the document window.
- To show the workspace surrounding the Stage, or to view elements in a scene that are partly or completely outside of the Stage area, select View > Pasteboard. The pasteboard appears in light gray. For example, to have a bird fly into a frame, initially position the bird outside of the Stage in the pasteboard and animate it into the Stage area.

Move the view of the Stage

When the Stage is magnified, you may not be able to see all of it. To change the view without having to change the magnification, use the Hand tool to move the Stage.

- ❖ In the Tools panel, select the Hand tool and drag the Stage. To temporarily switch between another tool and the Hand tool, hold down the Spacebar and click the tool in the Tools panel.

Use rulers

When rulers show, they appear along the top and left sides of the document. You can change the unit of measure used in the rulers from the default of pixels to another unit. When you move an element on the Stage with the rulers displayed, lines indicating the element's dimensions appear on the rulers.

- To show or hide rulers, select View > Rulers.
- To specify the rulers' unit of measure for a document, select Modify > Document, and select a unit from the Ruler Units menu.

See also

[“Snapping art into position”](#) on page 122

Use guides

When rulers show (View > Rulers), you can drag horizontal and vertical guides from the rulers onto the Stage.

When you create nested timelines, draggable guides appear on the Stage only when the Timeline in which they were created is active.

To create custom guides or irregular guides, use guide layers.

- To display or hide the drawing guides, select View > Guides > Show Guides.
Note: If the grid is visible and Snap to Grid is turned on when you create guides, guides snap to the grid.
- To turn snapping to guides on or off, select View > Snapping > Snap to Guides.
Note: Snapping to guides takes precedence over snapping to the grid in places where guides fall between grid lines.
- To move a guide, click anywhere on the ruler with the Selection tool and drag the guide to the desired place on the Stage.
- To remove a guide, use the Selection tool with guides unlocked to drag the guide to the horizontal or vertical ruler.
- To lock guides, select View > Guides > Lock Guides or use the Lock Guides option in the Edit Guides (View > Guides > Edit Guides) dialog box.
- To clear guides, select View > Guides > Clear Guides. If you are in document-editing mode, all guides in the document are cleared. If you are in symbol-editing mode, only guides used in symbols are cleared.

See also

[“Guide layers”](#) on page 177

Set guide preferences

1 Select View > Guides > Edit Guides and do any of the following:

- To set Color, click the triangle in the color box and select a guide line color from the palette. The default guide color is green.

- To display or hide guides, select or deselect Show Guides.
 - To turn snapping to guides on or off, select or deselect Snap To Guides.
 - Select or deselect Lock Guides.
 - To set Snap Accuracy, select an option from the pop-up menu.
 - To remove all guides, click Clear All. Clear All removes all guides from the current scene.
 - To save the current settings as the default, click Save Default.
- 2 Click OK.

Use the grid

The grid appears in a document as a set of lines behind the artwork in all scenes.

Display or hide the drawing grid

- ❖ Do one of the following:
 - Select View > Grid > Show Grid.
 - Press Control+" (quote) (Windows) or Command+" (quote) (Macintosh).

Turn snapping to grid lines on or off

- ❖ Select View > Snapping > Snap to Grid.

Set grid preferences

- 1 Select View > Grid > Edit Grid and select from the options.
- 2 To save the current settings as the default, click Save Default.

About the main toolbar and edit bar

The menu bar at the top of the application window contains menus with commands for controlling functionality.

The edit bar, at the top of the Stage, contains controls and information for editing scenes and symbols, and for changing the magnification level of the Stage.

See also

[“Symbols, instances, and library assets”](#) on page 147

[“Working with scenes”](#) on page 178

Tools panel overview

The tools in the Tools panel let you draw, paint, select, and modify artwork, as well as change the view of the Stage. The Tools panel is divided into four sections:

- The tools area contains drawing, painting, and selection tools.
- The view area contains tools for zooming and panning in the application window.
- The colors area contains modifiers for stroke and fill colors.
- The options area contains modifiers for the currently selected tool. Modifiers affect the tool’s painting or editing operations.

To specify which tools to display in the authoring environment, use the Customize Tools Panel dialog box.

See also

[“Creating and Editing Artwork”](#) on page 91

[“Selecting objects”](#) on page 118

Use the Tools panel

To show or hide the Tools panel, select Window > Tools.

Select tools

❖ Do one of the following:

- Click the tool in the Tools panel. Depending on the tool you select, a set of modifiers might appear in the options area at the bottom of the Tools panel.
- Press the tool’s keyboard shortcut. To view the keyboard shortcuts, select Edit > Keyboard Shortcuts (Windows) or Flash > Keyboard Shortcuts (Macintosh). On the Macintosh, you might need to move the mouse to see the new pointer appear.
- To select a tool located in the pop-up menu for a visible tool such as the Rectangle tool, press the icon of the visible tool and select another tool from the pop-up menu.

Customize the Tools panel

To specify which tools appear in the authoring environment, use the Customize Tools Panel dialog box to add or remove tools from the Tools panel.

When more than one tool appears in a location, the top tool in the group (the most recently used) appears with an arrow in the lower-right corner of its icon. This arrow indicates that additional tools are present in a pop-up menu. The same keyboard shortcut functions for all tools in the pop-up menu. When you press and hold the mouse button on the icon, the other tools in the group appear in a pop-up menu.

1 To show the Customize Tools Panel dialog box, do one of the following:

- (Windows) Select Edit > Customize Tools panel.
- (Macintosh) Select Flash > Customize Tools panel.

The Available Tools menu indicates the tools that are currently available. The Current Selection menu indicates the tools currently assigned to the selected location in the Tools panel.

- 2 To browse through the tools to specify the location to assign to another tool, click a tool in the image of the Tools panel or use the arrows.
- 3 To add a tool to the selected location, select the tool in the Available Tools list and click Add. You can assign a tool to more than one location.
- 4 To remove a tool from the selected location, select the tool in the Current Selection scroll list and click Remove.
- 5 To restore the default Tools Panel layout, click Restore Default in the Customize Tools Panel dialog box.
- 6 Click OK to apply your changes and close the Customize Tools Panel dialog box.

Use context menus

Context menus contain commands relevant to the current selection. For example, when you select a frame in the Timeline window, the context menu contains commands for creating, deleting, and modifying frames and keyframes. Context menus exist for many items and controls in many locations, including on the Stage, in the Timeline, in the Library panel, and in the Actions panel.

- ❖ Right-click (Windows) or Control-click (Macintosh) an item.

The Timeline

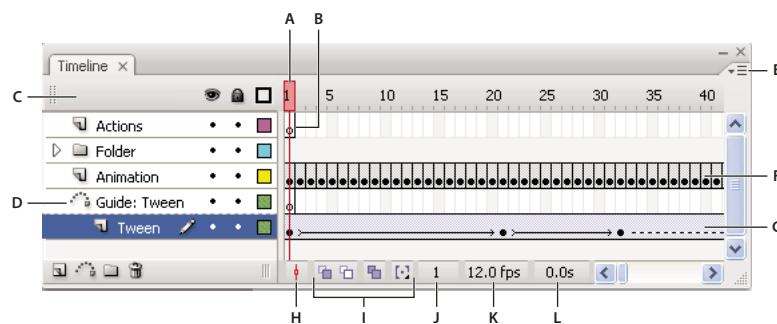
About the Timeline

The Timeline organizes and controls a document's content over time in layers and frames. Like films, Flash documents divide lengths of time into frames. Layers are like multiple film strips stacked on top of one another, each containing a different image that appears on the Stage. The major components of the Timeline are layers, frames, and the playhead.

Layers in a document are listed in a column on the left side of the Timeline. Frames contained in each layer appear in a row to the right of the layer name. The Timeline header at the top of the Timeline indicates frame numbers. The playhead indicates the current frame displayed on the Stage. As a document plays, the playhead moves from left to right through the Timeline.

The Timeline status displayed at the bottom of the Timeline indicates the selected frame number, the current frame rate, and the elapsed time to the current frame.

Note: When an animation is played, the actual frame rate is displayed; this may differ from the document's frame rate setting if the computer can't calculate and display the animation quickly enough.



Parts of the Timeline

- A. Playhead B. Empty keyframe C. Timeline header D. Guide layer icon E. Frame View pop-up menu F. Frame-by-frame animation G. Tweened animation H. Scroll To Playhead button I. Onion-skinning buttons J. Current Frame indicator K. Frame Rate indicator L. Elapsed Time indicator

The Timeline shows where animation occurs in a document, including frame-by-frame animation, tweened animation, and motion paths.

Controls in the layers section of the Timeline let you hide, show, lock, or unlock layers, as well as display layer contents as outlines. You can drag frames to a new location on the same layer or to a different layer.

See also

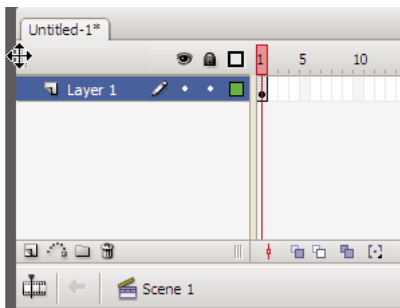
“[Working with timelines](#)” on page 172

“[Motion tweens](#)” on page 187

Change the appearance of the Timeline

By default, the Timeline appears below the main document window. To change its position, detach the Timeline from the document window and float it in its own window or dock it to any other panel you choose. You can also hide the Timeline.

To change the number of layers and frames that are visible, resize the Timeline. To view additional layers when the Timeline contains more layers than can be displayed, use the scroll bars on the right side of the Timeline.

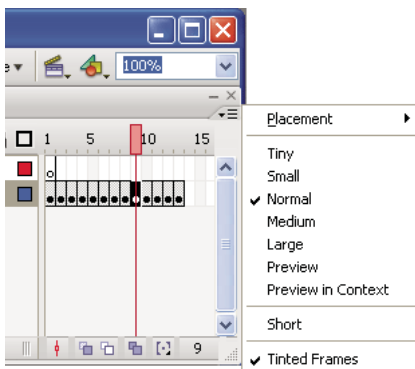


Dragging the Timeline

- To move the Timeline when it is docked to the document window, drag the title bar tab at the upper-left corner of the Timeline.
- To dock an undocked Timeline to the application window, drag the title bar tab to the top or bottom of the document window.
- To dock an undocked Timeline to other panels, drag the Timeline title bar tab to the location you choose. To prevent the Timeline from docking to other panels, press Control while you drag. A blue bar appears to indicate where the Timeline will dock.
- To lengthen or shorten layer name fields in the Timeline panel, drag the bar separating the layer names and the frames portions of the Timeline.

Change the display of frames in the Timeline

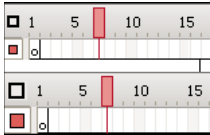
1 To display the Frame View pop-up menu, click Frame View in the upper-right corner of the Timeline.



Frame View pop-up menu.

2 Select from the following options:

- To change the width of frame cells, select Tiny, Small, Normal, Medium, or Large. (The Large frame-width setting is useful for viewing the details of sound waveforms.)
- To decrease the height of frame cell rows, select Short.



Short and Normal frame view options.

- To turn the tinting of frame sequences on or off, select Tinted Frames.
- To display thumbnails of the content of each frame scaled to fit the Timeline frames, select Preview. This can cause the apparent content size to vary and requires extra screen space.
- To display thumbnails of each full frame (including empty space), select Preview In Context. This is useful for viewing the way elements move in their frames over the course of the animation, but previews are generally smaller than with the Preview option.

Change layer height in the Timeline

1 Do one of the following:

- Double-click the layer's icon (the icon to the left of the layer name) in the Timeline.
- Right-click (Windows) or Control-click (Macintosh) the layer name and select Properties from the context menu.
- Select the layer in the Timeline and select Modify > Timeline > Layer Properties.

2 In the Layer Properties dialog box, select an option for Layer Height and click OK.

Resize the Timeline

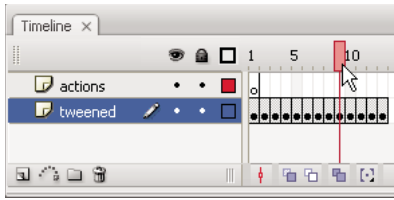
- If the Timeline is docked to the main application window, drag the bar separating the Timeline from the Stage area.
- If the Timeline is not docked to the main application window, drag the lower-right corner (Windows) or the size box in the lower-right corner (Macintosh).

Move the playhead

The playhead moves through the timeline as a document plays to indicate the current frame displayed on the Stage. The Timeline header shows the frame numbers of the animation. To display a frame on the Stage, move the playhead to the frame in the Timeline.

To display a specific frame when you're working with a large number of frames that can't all be displayed in the Timeline at once, move the playhead along the Timeline.

- To go to a frame, click the frame's location in the Timeline header, or drag the playhead to the desired position.
- To center the Timeline on the current frame, click the Scroll To Playhead button at the bottom of the Timeline.



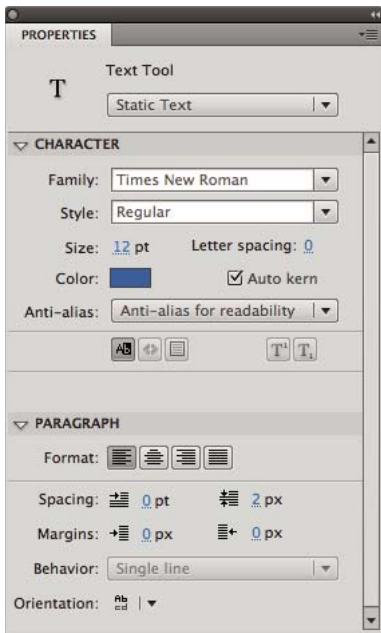
Moving the playhead

Using Flash authoring panels

About the Property inspector

The Property inspector provides easy access to the most commonly used attributes of the current selection, either on the Stage or in the Timeline. You can make changes to the object or document attributes in the Property inspector without accessing the menus or panels that also control these attributes.

Depending on what is currently selected, the Property inspector displays information and settings for the current document, text, symbol, shape, bitmap, video, group, frame, or tool. When two or more different types of objects are selected, the Property inspector displays the total number of objects selected.

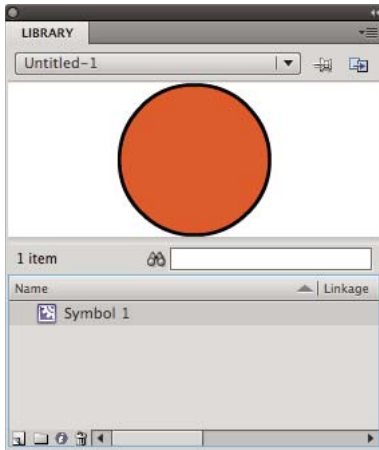


The Property inspector showing the properties for the Text tool.

To display the Property inspector, Select Window > Properties, or press Control+F3 (Windows) or Command+F3 (Macintosh).

About the Library panel

The Library panel (Window > Library) is where you store and organize symbols created in Flash, as well as imported files, including bitmap graphics, sound files, and video clips. The Library panel lets you organize library items in folders, see how often an item is used in a document, and sort items by name, type, date, use count, or ActionScript® linkage identifier. You can also search the Library panel with the search field and set properties on most multiple-object selections.



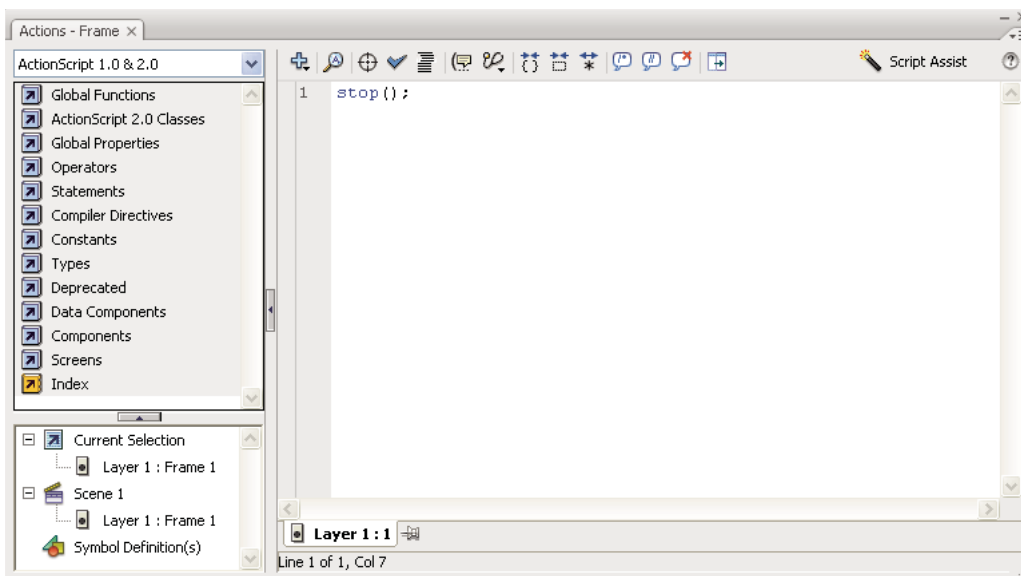
The Library panel showing a movie clip symbol.

See also

“[Managing assets with the library](#)” on page 157

About the Actions panel

The Actions panel lets you create and edit ActionScript code for an object or frame. Selecting a frame, button, or movie clip instance makes the Actions panel active. The Actions panel title changes to Button Actions, Movie Clip Actions, or Frame Actions, depending on what is selected.



The Actions panel showing a stop() action in a frame.

To display the Actions panel, select Window > Actions or press F9.

See also

[“Actions panel overview”](#) on page 332

[“Script window overview”](#) on page 334

Use the Movie Explorer

The Movie Explorer lets you view and organize the contents of a document and select elements in the document for modification. It contains a display list of currently used elements, arranged in a navigable hierarchical tree.

Use the Movie Explorer to perform the following actions:

- Filter which categories of items in the document appear in the Movie Explorer.
- Display the selected categories as scenes, symbol definitions, or both.
- Expand and collapse the navigation tree.
- Search for an element in a document by name.
- Familiarize yourself with the structure of a Flash document that another developer created.
- Find all the instances of a particular symbol or action.
- Print the navigable display list that appears in the Movie Explorer.

The Movie Explorer has a Panel menu and a context menu with options for performing operations on selected items or modifying the Movie Explorer display. A check mark with a triangle below it in the Movie Explorer panel indicates the Panel menu.

Note: The Movie Explorer has slightly different functionality when you are working with screens.

See also

[“Working with screens”](#) on page 316

View the Movie Explorer

- ❖ Select Window > Movie Explorer.

Filter the categories of items that appear in the Movie Explorer

- To show text, symbols, ActionScript, imported files, or frames and layers, click one or more of the filtering buttons to the right of the Show option. To customize which items to show, click the Customize button. Select options in the Show area of the Movie Explorer Settings dialog box to view those elements.
- To show items in scenes, select Show Movie Elements from the Movie Explorer Panel menu.
- To show information about symbols, select Show Symbol Definitions from the Movie Explorer Panel menu.

Note: The Movie Elements option and the Symbol Definitions option can be active at the same time.

Search for an item using the Find box

- ❖ In the Find box, enter the item name, font name, ActionScript string, or frame number. The Find feature searches all items that appear in the Movie Explorer.

Select an item in the Movie Explorer

- ❖ Click the item in the navigation tree. Shift-click to select more than one item.

The full path for the selected item appears at the bottom of the Movie Explorer. Selecting a scene in the Movie Explorer shows the first frame of that scene on the Stage. Selecting an element in the Movie Explorer selects that element on the Stage if the layer containing the element is not locked.

Use the Movie Explorer Panel menu or context menu commands

1 Do one of the following:

- To view the Panel menu, click the Panel menu control in the Movie Explorer panel.
- To view the context menu, right-click (Windows) or Control-click (Macintosh) an item in the Movie Explorer navigation tree.

2 Select an option from the menu:

Go To Location Jumps to the selected layer, scene, or frame in the document.

Go To Symbol Definition Jumps to the symbol definition for a symbol that is selected in the Movie Elements area of the Movie Explorer. The symbol definition lists all the files associated with the symbol. (The Show Symbol Definitions option must be selected. See its definition in this list.)

Select Symbol Instances Jumps to the scene containing instances of a symbol that is selected in the Symbol Definitions area of the Movie Explorer. (The Show Movie Elements option must be selected.)

Show In Library Highlights the selected symbol in the document's library. (Flash opens the Library panel if it is not already visible.)

Rename Lets you enter a new name for a selected element.

Edit In Place Lets you edit a selected symbol on the Stage.

Edit In New Window Lets you edit a selected symbol in a new window.

Show Movie Elements Shows the elements in your document organized into scenes.

Show Symbol Definitions Shows all the elements associated with a symbol.

Copy All Text To Clipboard Copies selected text to the clipboard. For spell checking or other editing, paste the text into an external text editor.

Cut, Copy, Paste, And Clear Performs these common functions on a selected element. Modifying an item in the display list modifies the corresponding item in the document.

Expand Branch Expands the navigation tree at the selected element.

Collapse Branch Collapses the navigation tree at the selected element.

Collapse Others Collapses the branches in the navigation tree that do not contain the selected element.

Print Prints the hierarchical display list that appears in the Movie Explorer.

About Flash components and the Components panel

A component in Flash is a reusable, packaged module that adds a particular capability to a Flash document. Components can include graphics as well as code, so they're pre-built functionality that you can easily include in your Flash projects. For example, a component can be a radio button, a dialog box, a preload bar, or even something that has no graphics at all, such as a timer, a server connection utility, or a custom XML parser.

If you are less experienced with writing ActionScript, you can add components to a document, set their parameters in the Property inspector or Component inspector, and use the Behaviors panel to handle their events. For example, you could attach a Go To Web Page behavior to a Button component that opens a URL in a web browser when the button is clicked without writing any ActionScript code.

If you are a programmer who wants to create more robust applications, you can create components dynamically, use ActionScript to set properties and call methods at run time, and use the event listener model to handle events.

Insert a component using the Component panel

When you first add a component to a document, Flash imports it as a movie clip into the Library panel. You can also drag a component from the Components panel directly to the Library panel and then add an instance of it to the Stage. In any case, you must add a component to the library before you can access its class elements.

- 1 Select Window > Component panel.
- 2 Select an instance of a component in the Component panel, and drag it to the Stage or Library panel. After a component is added to the library, you can drag multiple instances to the Stage.
- 3 Configure the component as needed using either the Property inspector or the Components inspector. For information on the parameters the component uses, refer to the appropriate component documentation for the version of ActionScript you are using in the Flash document.

Enter parameters for a component using the Component inspector

- 1 Select Window > Component Inspector.
- 2 Select an instance of a component on the Stage.
- 3 Click the Parameters tab and enter values for any of the listed parameters.

About the Web Services panel

You can view a list of web services, refresh web services, and add or remove web services in the Web Services panel (Window > Other Panels > Web Services). When you add a web service to the Web Services panel, the web service is then available to any application you create.

You can use the Web Services panel to refresh all your web services at once by clicking the Refresh Web Services button. If you are not using the Stage but instead are writing ActionScript code for the connectivity layer of your application, you can use the Web Services panel to manage your web services.

For detailed information about using the web services panel, see www.adobe.com/go/learn_fl_web_services.

Undo, redo, and history

Undo, Redo, and Repeat commands

To undo or redo actions on individual objects, or all objects within the current document, specify either object-level or document-level Undo and Redo commands (Edit > Undo or Edit Redo). The default behavior is document-level Undo and Redo.

You cannot undo some actions when using object-level Undo. Among these are entering and exiting Edit mode; selecting, editing, and moving library items; and creating, deleting, and moving scenes.

- To remove deleted items from a document after using the Undo command, use the Save And Compact command.

- To reapply a step to the same object or to a different object, use the Repeat command. For example, if you move a shape named shape_A, select Edit > Repeat to move the shape again, or select another shape, shape_B, and select Edit > Repeat to move the second shape by the same amount.

By default, Flash supports 100 levels of undo for the Undo menu command. Select the number of undo and redo levels, from 2 to 9999, in Flash Preferences.

By default, when you undo a step by using Edit > Undo or the History panel, the file size of the document does not change, even if you delete an item in the document. For example, if you import a video file into a document, and undo the import, the file size of the document still includes the size of the video file. Any items that you delete from a document when performing an Undo command are preserved in order to be able to restore the items with a Redo command. To permanently remove the deleted items from the document, and reduce the document file size, select File > Save And Compact.

See also

[“Set preferences in Flash”](#) on page 37

[“Automating tasks with the Commands menu”](#) on page 32

Using the History panel

The History panel (Window > Other Panels > History) shows a list of the steps you’ve performed in the active document since you created or opened that document, up to a specified maximum number of steps. (The History panel doesn’t show steps you’ve performed in other documents.) The slider in the History panel initially points to the last step that you performed.

- To undo or redo individual steps or multiple steps at once, use the History panel. Apply steps from the History panel to the same object or to a different object in the document. However, you cannot rearrange the order of steps in the History panel. The History panel is a record of steps in the order in which they are performed.

***Note:** If you undo a step or a series of steps and then do something new in the document, you can no longer redo the steps in the History panel; they disappear from the panel.*

- To remove deleted items from a document after you undo a step in the History panel, use the Save And Compact command.

By default, Flash supports 100 levels of undo for the History panel. Select the number of undo and redo levels, from 2 to 9999, in Flash Preferences.

- To erase the history list for the current document, clear the History panel. After clearing the history list, you cannot undo the steps that are cleared. Clearing the history list does not undo steps; it removes the record of those steps from the current document’s memory.

Closing a document clears its history. To use steps from a document after that document is closed, copy the steps with the Copy Steps command or save the steps as a command.

See also

[“Set preferences in Flash”](#) on page 37

[“Automating tasks with the Commands menu”](#) on page 32

Undo steps with the History panel

When you undo a step, the step is dimmed in the History panel.

- To undo the last step performed, drag the History panel slider up one step in the list.
- To undo multiple steps at once, drag the slider to point to any step, or click to the left of a step along the path of the slider. The slider scrolls automatically to that step, undoing all subsequent steps as it scrolls.

Note: Scrolling to a step (and selecting the subsequent steps) is different from selecting an individual step. To scroll to a step, click to the left of the step.

Replay steps with the History panel

When you replay steps with the History panel, the steps that play are the steps that are selected (highlighted) in the History panel, not necessarily the step currently indicated by the slider.

Apply steps in the History panel to any selected object in the document.

Replay one step

- ❖ In the History panel, select a step and click the Replay button.

Replay a series of adjacent steps

1 Select steps in the History panel by doing one of the following:

- Drag from one step to another. (Don't drag the slider; drag from the text label of one step to the text label of another step.)
- Select the first step, then Shift-click the last step; or select the last step and Shift-click the first step.

2 Click Replay. The steps replay in order, and a new step, labeled Replay Steps, appears in the History panel.

Replay nonadjacent steps

1 Select a step in the History panel, and Control-click (Windows) or Command-click (Macintosh) other steps. To deselect a selected step, Control-click or Command-click.

2 Click Replay.

Copy and paste steps between documents

Each open document has its own history of steps. To copy steps from one document and paste them into another, use the Copy Steps command in the History panel options menu. If you copy steps into a text editor, the steps are pasted as JavaScript™ code.

1 In the document containing the steps to reuse, select the steps in the History panel.

2 In the History panel options menu, select Copy Steps.

3 Open the document to paste the steps into.

4 Select an object to apply the steps to.

5 Select Edit > Paste to paste the steps. The steps play back as they're pasted into the document's History panel. The History panel shows them as only one step, called Paste Steps.

Automating tasks with the Commands menu

Create and manage commands

To repeat the same task, create a command in the Commands menu from steps in the History panel and reuse the command. Steps replay exactly as they were originally performed. You can't modify the steps as you replay them.

To use steps the next time you start Flash, create and save a command. Saved commands are retained permanently, unless you delete them. Steps that you copy using the History panel Copy Steps command are discarded when you copy something else.

Create a command from selected steps in the History panel. Rename or delete commands in the Manage Saved Commands dialog box.

See also

[“Copy and paste steps between documents”](#) on page 31

Create a command

- 1 Select a step or set of steps in the History panel.
- 2 Select Save As Command from the History panel options menu.
- 3 Enter a name for the command and click OK. The command appears in the Commands menu.

Note: The command is saved as a JavaScript file (with the extension .jsfl) in your Commands folder. This folder is in the following locations: Windows 2000 or Windows XP: boot drive\Documents and Settings\

Edit the names of commands in the Commands menu

- 1 Select Commands > Manage Saved Commands.
- 2 Select a command to rename.
- 3 Click the Rename button.
- 4 Enter the new name and click OK.

Delete a name from the Commands menu

- 1 Select Commands > Manage Saved Commands, and select a command.
- 2 Click Delete, click Yes in the warning dialog box, and click OK.

Run commands

- To use a saved command, select the command from the Commands menu.
- To run a JavaScript or Flash JavaScript command, select Commands > Run Command, navigate to the script to run, and click Open.

Get more commands

Use the Get More Commands option in the Commands menu to link to the Flash Exchange website at www.adobe.com/go/flash_exchange and download commands that other Flash users have posted. For more information on the commands posted there, see the Flash Exchange website.

- 1 Make sure you are connected to the Internet.
- 2 Select Commands > Get More Commands.

Steps that can't be used in commands

Some tasks can't be saved as commands or repeated using the Edit > Repeat menu item. These commands can be undone and redone, but they cannot be repeated.

Examples of actions that can't be saved as commands or repeated include selecting a frame or modifying a document size. If you attempt to save an unrepeatable action as a command, the command is not saved.

Accessibility in the Flash workspace

The Adobe® Flash® CS4 Professional workspace is keyboard-accessible for users who may have difficulty using a mouse.

About accessibility support

Accessibility support in the authoring environment provides keyboard shortcuts for navigating and using interface controls, including panels, the Property inspector, dialog boxes, the Stage, and objects on the Stage, so that you can work with these interface elements without using the mouse.

Note: Certain keyboard controls and authoring environment accessibility features are available only in Windows.

To customize the keyboard shortcuts for accessibility in the authoring environment, use the Workspace Accessibility Commands section of the Keyboard Shortcuts dialog box.

See also

[“Customize keyboard shortcut sets”](#) on page 41

About Flash authoring accessibility on the Macintosh

Accessibility for the authoring environment on the Macintosh has the following limitations:

- The Panel Focus keyboard shortcut (Command+Option+Tab) is not supported for the Property inspector.
- The Panel Control Focus keyboard shortcut (Tab) is supported only for the Timeline, not for other panels or the Property inspector.

Select panels or the Property inspector with keyboard shortcuts

To select a panel or the Property inspector (also referred to as applying *focus* to the panel or Property inspector), use the keyboard shortcut Control+F6 (Windows) or Command+F6 (Macintosh).

Apply focus to a panel or the Property inspector only when the panel or Property inspector is visible in the application window. The panel can be expanded or collapsed.

When you use the keyboard shortcut to select panels, focus is applied to panels using the following criteria:

- Docked panels are given focus first.
- If the Timeline is showing and docked, the Timeline is given focus the first time you press Control+F6 (Windows) or Command+F6 (Macintosh).
- If the Timeline is not showing and docked, or if you press the keyboard shortcut again, focus moves to the rightmost and highest docked panel. Pressing the keyboard shortcut repeatedly then moves the focus through the other docked panels, from right to left and from top to bottom of the workspace.
- If you move the focus through all the docked panels, or if no docked panels are showing, focus moves to the rightmost and highest floating panel. Pressing the keyboard shortcut repeatedly then moves the focus through the other floating panels, from right to left and from top to bottom of the workspace.

Use keyboard shortcuts to select or deselect, expand, or collapse panels or the Property inspector

- To move the focus through the panels currently displayed in the workspace, press Control+F6 (Windows) or Command+F6 (Macintosh). A dotted line appears around the title of the currently focused panel.
- To move the focus to the previously selected panel, press Control+Shift+F6 (Windows) or Command+Shift+F6 (Macintosh).
- To deselect a panel, press Escape, or move, dock, or undock the panel.
- To move the focus to the panel above or below the current panel in a panel group, press Up Arrow or Down Arrow.
- To hide all panels and the Property inspector, press F4. To display all panels and the Property inspector, press F4 again.

Use keyboard shortcuts to expand or collapse panels or the Property inspector

- 1 Press Control+F6 (Windows) or Command+F6 (Macintosh) until the panel to expand or collapse has focus. A dotted line appears around the title of the currently focused panel.
- 2 To expand or collapse the currently selected panel, press the Spacebar.

Select controls in a panel or the Property inspector using keyboard shortcuts

To move the focus through the panel controls when a panel or the Property inspector has the current focus, use the Tab key. To activate the control that has the current focus, use the Spacebar (that is, pressing Spacebar is equivalent to clicking a control in the panel).

When you use the keyboard shortcut for panel controls, focus is applied to a control and the control is activated using the following criteria:

- To select a control in the panel with the Tab key, the panel with the current focus must be expanded. If the panel is collapsed, pressing Tab has no effect.
- When the panel with the current focus is expanded, pressing Tab the first time moves the focus to the panel's Panel menu.
- To move the focus between the Panel menu and the panel title bar, use Right Arrow and Left Arrow.
- If the focus is on the Panel menu, press Tab again to move the focus through the other controls in the panel. Pressing Tab again does not return the focus to the Panel menu.
- To display the Panel menu items when the Panel menu has the focus, press Enter (Windows only).

- To move the focus between the Panel menus of the panels in the group in panels that are grouped, use Up Arrow and Down Arrow.
- You can move the focus to a panel control only if the control is active. If a control is dimmed (inactive), you cannot apply focus to the control.

Move the focus from a panel title bar to a panel options menu

❖ Do one of the following:

- Press Tab.
- Press Right Arrow. To return the focus to the panel title bar, press Left Arrow or Shift+Tab.
- To move the focus to the Panel menu of the panel immediately above the panel with the current focus if the panel is in a group, press Up Arrow. To move the focus to the Panel menu of the panel immediately below the panel with the current focus, press Down Arrow.

Move the focus through the items in the Panel menu of a panel

- 1 To display the Panel menu items when the focus currently applied to the Panel menu, press the Spacebar.
- 2 To move through the items in the Panel menu, press Down Arrow.
- 3 To activate the currently selected Panel menu item, press Enter (Windows) or Return (Macintosh).

Move the focus through the controls in a panel

- 1 Press Tab when the focus is currently applied to the Panel menu. To move the focus through the controls in the panel, press Tab repeatedly.
- 2 To activate the currently selected panel control, press Enter (Windows only).

Navigate dialog box controls using keyboard shortcuts (Windows only)

- To move through the controls in the dialog box, press Tab.
- To move through the controls within one section of a dialog box, press Up Arrow and Down Arrow.
- To activate the button (equivalent to clicking the button), when the focus is applied to a dialog box control button, press Enter.
- To apply the current settings and close the dialog box (equivalent to clicking OK), when the focus is not applied to any dialog box control button, press Enter.
- To close the dialog box without applying the changes (equivalent to clicking Cancel), press Escape.
- To view the Help content for the dialog box (equivalent to clicking Help), when the focus is applied to the Help button, press Enter or Spacebar.

Select the Stage or objects on the Stage using keyboard shortcuts

Selecting the Stage with a keyboard shortcut is equivalent to clicking on the Stage. Any other element currently selected becomes deselected when the Stage is selected.

After the Stage is selected, use the Tab key to navigate through all objects on all layers, one at a time. You can select instances (including graphic symbols, buttons, movie clips, bitmaps, videos, or sounds), groups, or boxes. You cannot select shapes (such as rectangles) unless those shapes are instances of symbols. You cannot select more than one object at a time using keyboard shortcuts.

To select the Stage or objects on the Stage, use the following techniques:

- To select the Stage, press Control+Alt+Home (Windows) or Command+Option+Home (Macintosh).
- To select an object on the Stage, with the Stage selected, press Tab.
- To select the previous object when an object is currently selected, press Shift+Tab.
- To select the first object that was created on the active frame in the active layer, press Tab. When the last object on the top layer is selected, press Tab to move to the next layer beneath it and select the first object there, and so on.
- When the last object on the last layer is selected, press Tab to move to the next frame and select the first object on the top layer there.
- Objects on layers that are hidden or locked cannot be selected with the Tab key.

Note: If you are currently typing text in a box, you cannot select an object using the keyboard focus. You must first change the focus to the Stage and then select an object.

Navigate tree structures using keyboard shortcuts

To navigate *tree structures*, the hierarchical displays of file structures in certain Flash panels, use keyboard shortcuts.

- To expand a collapsed folder, select the folder and press Right Arrow.
- To collapse an expanded folder, select the folder and press Left Arrow.
- To move to the parent folder of an expanded folder, press Left Arrow.
- To move to the child folder of an expanded folder, press Right Arrow.

Work with library items using keyboard shortcuts

- 1 To copy or paste a selected library item, press Control+X (Windows) or Command+X (Macintosh) to cut the item, or press Control+C (Windows) or Command+C (Macintosh) to copy the item.
- 2 To paste a cut or copied item, click the Stage or in another library to set the insertion point, and press Control+V (Windows) or Command+V (Macintosh) to paste in the center of the Stage; or press Control+Shift+C (Windows) or Command+Shift+C (Macintosh) to paste in place (in the same location as the original).

To cut, copy, and paste items, use the following techniques:

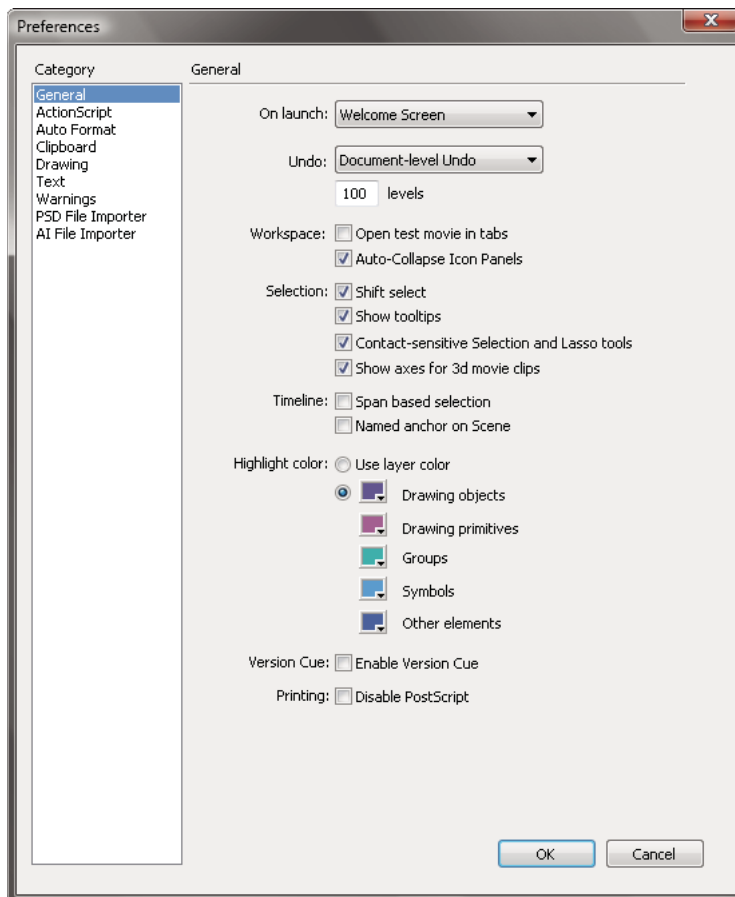
- Cut or copy one item or multiple items.
- Cut or copy an item from the Library panel and paste it onto the Stage or into another library, or paste a folder into another library.
- You cannot paste a shape from the Stage into the library.
- You cannot paste a library item into a common library, because common libraries cannot be modified. However, you can create a common library.
- When you paste a library item onto the Stage, the item is centered.
- If you paste a folder, each item in the folder is included.
- To paste a library item into a folder in the destination library, click the folder before pasting.
- You can paste a library item into a different location in the same library where it originated.
- If you attempt to paste a library item into a location containing another item by the same name, select whether to replace the existing item.

See also

[“Work with common libraries”](#) on page 160

Set preferences in Flash

You can set preferences for general application operations, editing operations, and clipboard operations.



The General category in the Preferences dialog box.

See also

[“Drawing preferences”](#) on page 96

[“Change the appearance of the Timeline”](#) on page 23

[“About the Timeline”](#) on page 22

[“Managing documents”](#) on page 43

[“Substituting missing fonts”](#) on page 253

[“Pen tool preferences”](#) on page 107

[“Illustrator object import options”](#) on page 76

[“Photoshop file importer preferences”](#) on page 84

Set preferences

- 1 Select Edit > Preferences (Windows) or Flash > Preferences (Macintosh).
- 2 Make a selection in the Category list and select from the respective options.

Set AutoFormat preferences for ActionScript

- ❖ Select any of the options. To see the effect of each selection, look in the Preview pane.

Set text preferences

- For Font Mapping Default, select a font to use when substituting missing fonts in documents you open in Flash.
- For Vertical Text options, select Default Text Orientation (deselected by default).
- To reverse the default text display direction, select Right To Left Text Flow (deselected by default).
- To turn off kerning for vertical text, select No Kerning (deselected by default). Turning off kerning is useful to improve spacing for some fonts that use kerning tables.
- For Input Method, select the appropriate language.

Set warning preferences

- To receive a warning when you try to save documents with content that is specific to the Adobe® Flash® CS4 Professional authoring tool as a Flash 8 file, select Warn On Save For Adobe Flash 8 and CS3 Compatibility (default).
- To receive a warning if the URL for a document changed since the last time you opened and edited it, select Warn On URL Changes In Launch And Edit.
- To place a red X over any Generator objects as a reminder that Generator objects are not supported in Flash 8, select Warn On Reading Generator Content.
- To receive an alert when Flash inserts frames in your document to accommodate audio or video files that you import, select Warn On Inserting Frames When Importing Content.
- To receive an alert when selecting Default Encoding that could potentially lead to data loss or character corruption, select Warn On Encoding Conflicts When Exporting ActionScript Files. (For example, if you create a file with English, Japanese, and Korean characters and select Default Encoding on an English system, the Japanese and Korean characters are corrupted.)
- To receive a warning when you attempt to edit a symbol with timeline effects applied to it, select Warn On Conversion Of Effect Graphic Objects.
- To receive a warning when you create a site in which the local root folder overlaps with another site, select Warn On Sites With Overlapped Root Folder.
- To receive a warning when you convert a symbol with a behavior attached to a symbol of a different type—for example, when you convert a movie clip to a button—select Warn On Behavior Symbol Conversion.
- To receive a warning when you convert a symbol to a symbol of a different type, select Warn On Symbol Conversion.
- To receive a warning when Flash converts a graphic object drawn in Object Drawing mode to a group, select Warn On Automatically Converting From Drawing Object To Group.
- To display warnings on controls for features not supported by the Flash Player version that the current FLA file is targeting in its publish settings, select Show Incompatibility Warnings On Feature Controls.

Set General preferences

On Launch Specify which document opens when you start the application.

Document- or Object-level undo Document-level undo maintains a single list of all your actions for the entire Flash document. Object-level undo maintains separate lists of your actions for each object in your document. Object-level lets you undo an action on one object without having to also undo actions on other objects that might have been modified more recently than the target object.

Undo levels To set the number of undo or redo levels, enter a value from 2 to 300. Undo levels require memory; the more undo levels you use, the more system memory is consumed. The default is 100.

Workspace To open a new document tab in the application window when you select Control > Test Movie, select Open Test Movie In Tabs. The default is to open the test movie in its own window. To have panels in icon mode auto collapse when you click outside them, select Auto-Collapse Icon Panels.

Selection To control how multiple elements are selected, select or deselect Shift Select. When Shift Select is off, clicking additional elements adds them to the current selection. When Shift Select is on, clicking additional elements deselects other elements unless you hold down Shift.

Show Tooltips Shows tooltips when the pointer pauses over a control. To hide the tooltips, deselect this option.

Contact Sensitive Selects objects when any part of them is included in the marquee rectangle when dragging with the Selection or Lasso tools. The default is that objects are only selected when the tool's marquee rectangle completely surrounds the object.

Show axes for 3D Displays an overlay of the X, Y, and Z axes on all 3D movie clips. This makes them easier to identify on the Stage.

Timeline To use span-based selection in the Timeline, rather than the default frame-based selection, select Span Based Selection.

Named Anchor On Scene Make the first frame of each scene in a document a named anchor. Named anchors let you use the Forward and Back buttons in a browser to jump from scene to scene.

Highlight Color To use the current layer's outline color, select a color from the panel, or select Use Layer Color.

Version Cue Select this option to enable Version Cue®.

Printing (Windows only) To disable PostScript® output when printing to a PostScript printer, select Disable PostScript. By default, this option is deselected. Select this option if you have problems printing to a PostScript printer; however, this option slows down printing.

Clipboard preferences

Bitmaps (Windows only)

To specify Color Depth and Resolution parameters for bitmaps copied to the clipboard, select their respective options.

To apply anti-aliasing, select Smooth.

To specify the amount of RAM that is used when placing a bitmap image on the Clipboard, enter a value in the Size Limit text field. Increase this value when working with large or high-resolution bitmap images.

Gradient Quality To specify the quality of gradient fills placed in the Windows metafile, select an option. Choosing a higher quality increases the time required to copy artwork. To specify gradient quality when pasting items to a location outside of Flash, use this setting. When you are pasting in Flash, the full gradient quality of the copied data is preserved regardless of the Gradients setting on the Clipboard.

PICT Settings (Macintosh only) To specify how PICT files are handled, use the following settings.

- **Type** To preserve data copied to the Clipboard as vector artwork, select Objects. To convert the copied artwork to a bitmap, select one of the bitmap formats.
- **Resolution** Enter a value.
- **Include PostScript** Select to include PostScript data.
- **Gradient Quality** To specify gradient quality in the PICT file, select an option. Choosing a higher quality increases the time required to copy artwork. To specify gradient quality when pasting items to a location outside of Flash, use the Gradients setting. When you are pasting in Flash, the full gradient quality of the copied data is preserved regardless of the Gradient setting.
- **FreeHand Text** To keep text editable in a pasted FreeHand® file, select Maintain Text As Blocks.

Keyboard shortcuts

Create and modify custom keyboard shortcuts

You can create and modify keyboard shortcuts in Flash.

Customize keyboard shortcuts

- 1 Select Edit > Keyboard Shortcuts (Windows) or Flash > Keyboard Shortcuts (Macintosh).


The Keyboard Shortcuts dialog box appears.

- 2 Use the following options to add, delete, or edit keyboard shortcuts:

Current Set Lets you choose a set of predetermined shortcuts (listed at the top of the menu), or any custom set you've defined.

Commands Lets you select a category of commands to edit (for example, menu commands). The command list displays the commands associated with the category you selected from the Commands pop-up menu, along with the assigned shortcuts. The Menu Commands category displays this list as a tree view that replicates the structure of the menus. The other categories list the commands by name (such as Quit Application), in a flat list.


Shortcuts Displays all shortcuts assigned to the selected command.


Add Shortcut  Adds a new shortcut to the current command. To add a new blank line to the Shortcuts box, click this button. To add a new keyboard shortcut for this command, enter a new key combination and click Change. Each command can have two different keyboard shortcuts; if two shortcuts are already assigned to a command, the Add Item button does nothing.


Remove Shortcut  Removes the selected shortcut from the list of shortcuts.


Press Key Displays the key combination you enter when you're adding or changing a shortcut.

Change Adds the key combination shown in the Press Key box to the list of shortcuts, or changes the selected shortcut to the specified key combination.

Duplicate Set  Duplicates the current set. Give the new set a name; the default name is the current set's name with the word *copy* appended to it.


Rename Set  Renames the current set.

Export Set As HTML  Saves the current set in an HTML table format for easy viewing and printing. Open the HTML file in your browser and print the shortcuts for easy reference.


Delete Set  Deletes a set. You cannot delete the active set.

3 Click OK.

Remove a shortcut from a command

- 1 From the Commands pop-up menu, select a command category, select a command from the Commands list, and select a shortcut.
- 2 Click Remove Shortcut .

Add a shortcut to a command

- 1 From the Commands pop-up menu, select a command category and select a command.
- 2 Prepare to add a shortcut by doing one of the following:
 - If fewer than two shortcuts are already assigned to the command, click Add Shortcut . A new blank line appears in the Shortcuts box, and the insertion point moves to the Press Key box.
 - If two shortcuts are already assigned to the command, select one of them to be replaced by the new shortcut, and click in the Press Key box.
- 3 Press a key combination.

Note: If a problem occurs with the key combination (for example, if the key combination is already assigned to another command), an explanatory message appears just below the Shortcuts box and you may be unable to add or edit the shortcut.

4 Click Change.

Edit an existing shortcut


- 1 From the Commands pop-up menu, select a command category, select a command from the Commands list, and select a shortcut to change.
- 2 Click in the Press Key box, enter a new key combination, and click Change.

Note: If a problem occurs with the key combination (for example, if the key combination is already assigned to another command), an explanatory message appears just below the Shortcuts box and you may be unable to add or edit the shortcut.

Customize keyboard shortcut sets

To match the shortcuts you use in other applications, or to streamline your workflow, select keyboard shortcuts. By default, Flash uses built-in keyboard shortcuts designed for the application. You can also select a built-in keyboard shortcut set from one of several graphics applications.


View or print the current set of keyboard shortcuts

- 1 Select Edit > Keyboard Shortcuts (Windows) or Flash > Keyboard Shortcuts (Macintosh).
- 2 In the Keyboard Shortcuts dialog box, select the shortcut set to view from the Current Set pop-up menu.
- 3 Click the Export Set As HTML button .
- 4 Select a name and location for the exported HTML file. The default file name is the name of the selected shortcut set.
- 5 Click Save.
- 6 Find the exported file in the folder you selected and open the file in a web browser.
- 7 To print the file, use the browser's Print command.


Select a keyboard shortcut set

- 1 Select Edit > Keyboard Shortcuts (Windows) or Flash > Keyboard Shortcuts (Macintosh).
- 2 In the Keyboard Shortcuts dialog box, select a shortcut set from the Current Set pop-up menu.

Create a keyboard shortcut set

- 1 Select a keyboard shortcut set and click the Duplicate Set button .
- 2 Enter a name for the new shortcut set and click OK.

Rename a custom keyboard shortcut set

- 1 In the Keyboard Shortcuts dialog box, select a shortcut set from the Current Set pop-up menu.
- 2 Click the Rename Set button , enter a new name, and click OK.


Add or remove a keyboard shortcut

- 1 Select Edit > Keyboard Shortcuts (Windows) or Flash > Keyboard Shortcuts (Macintosh) and select the set to modify.
- 2 From the Commands pop-up menu, select a category to view shortcuts for the selected category.
- 3 In the Commands list, select the command for which you want to add or remove a shortcut. An explanation of the selected command appears in the description area in the dialog box.
- 4 Do one of the following:
 - To add a shortcut, click the Add Shortcut (+) button.
 - To remove a shortcut, click the Remove Shortcut (-) button and proceed to step 6.
- 5 If you are adding a shortcut, enter the new shortcut key combination in the Press Key box.

Note: To enter the key combination, press the keys on the keyboard. You do not need to spell out key names, such as Control, Option, and so on.

- 6 Click Change.
- 7 Repeat this procedure to add or remove additional shortcuts, and click OK.

Delete a keyboard shortcut set

- 1 Select Edit > Keyboard Shortcuts (Windows) or Flash > Keyboard Shortcuts (Macintosh). In the Keyboard Shortcuts dialog box, click Delete Set .
- 2 In the Delete Set dialog box, select a shortcut set and click Delete.

Note: You cannot delete the keyboard shortcut sets built into Flash.

Chapter 3: Managing documents

When you create and save Adobe® Flash® CS4 Professional documents within the Flash authoring environment, the documents are in FLA file format. To display a document in Adobe® Flash® Player, you must publish or export the document as a SWF file.

You can add media assets to a Flash document and manage the assets in the library, and you can use the Movie Explorer to view and organize all the elements in a Flash document. The Undo and Redo commands, the History panel, and the Commands menu let you automate tasks in a document.

Working with Flash documents

About Flash files

In Flash, you can work with a variety of file types, each of which has a separate purpose:

- FLA files, the primary files you work with in Flash, contain the basic media, timeline, and script information for a Flash document. *Media objects* are the graphic, text, sound, and video objects that comprise the content of your Flash document. The *Timeline* is where you tell Flash when specific media objects should appear on the Stage. You can add *ActionScript®* code to Flash documents to more finely control their behavior and to make them respond to user interactions.
- SWF files, the compiled versions of FLA files, are the files you display in a web page. When you publish your FLA file, Flash creates a SWF file.
- AS files are ActionScript files—you can use these to keep some or all of your ActionScript code outside of your FLA files, which is helpful for code organization and for projects that have multiple people working on different parts of the Flash content.
- SWC files contain the reusable Flash components. Each SWC file contains a compiled movie clip, ActionScript code, and any other assets that the component requires.
- ASC files are files used to store ActionScript that will be executed on a computer running Flash Media Server. These files provide the ability to implement server-side logic that works in conjunction with ActionScript in a SWF file.
- JSFL files are JavaScript files that you can use to add new functionality to the Flash authoring tool.

For video tutorials about working with Flash files, see the following:

- www.adobe.com/go/vid0117
- www.adobe.com/go/vid0118

See also

“[About the Timeline](#)” on page 22

Create a new document

You can create a new document or open a previously saved document in Flash, and you can open a new window as you work. You can set properties for new or existing documents.

For a video tutorial, see Working with Flash Files at www.adobe.com/go/vid0117.

See also

“[Set preferences in Flash](#)” on page 37

“[Publishing and Exporting](#)” on page 379

Create a new document

- 1 Select File > New.
- 2 On the General tab, select Flash Document.

On Windows, you can create a new document of the same type as the last document created by clicking the New File button in the main toolbar.

Create a new document from a template

- 1 Select File > New.
- 2 Click the Templates tab.
- 3 Select a category from the Category list, select a document from the Category Items list, and click OK. You can select standard templates that come with Flash or a template you have already saved.

Open an existing document

- 1 Select File > Open.
- 2 In the Open dialog box, navigate to the file or enter the path to the file in the Go To box.
- 3 Click Open.

Open a new window for the current document

- ❖ Select Window > Duplicate Window.

Set properties for a new or existing document

- 1 With the document open, select Modify > Document.

The Document Properties dialog box appears.

- 2 For Frame Rate, enter the number of animation frames to appear every second.

For most computer-displayed animations, especially those playing from a website, 8 frames per second (fps) to 15 fps is sufficient. When you change the frame rate, the new frame rate becomes the default for new documents.

- 3 For Dimensions, set the Stage size:

- To specify the Stage size in pixels, enter values in the Width and Height boxes. The minimum size is 1 x 1 pixels; the maximum is 2880 x 2880 pixels.
- To set the Stage size so that there is equal space around the content on all sides, click the Contents option to the right of Match. To minimize document size, align all elements to the upper-left corner of the Stage, and then click Contents.
- To set the Stage size to the maximum available print area, click Printer. This area is determined by the paper size minus the current margin selected in the Margins area of the Page Setup dialog box (Windows) or the Print Margins dialog box (Macintosh).
- To set the Stage size to the default size, 550 x 400 pixels, click Default.

- 4 To set the background color of your document, click the triangle in the Background Color control and select a color from the palette.

- 5 To specify the unit of measure for rulers that you can display along the top and side of the application window, select an option from the Ruler Units menu in the lower left. (This setting also determines the units used in the Info panel.)
- 6 Do one of the following:
 - To make the new settings the default properties for the current document only, click OK.
 - To make the new settings the default properties for all new documents, click Make Default.

Change document properties using the Property inspector

- 1 Deselect all assets, then select the Selection tool.
- 2 In the Property inspector (Window > Properties), click the Edit button next to the Size property to display the Document Properties dialog box.
- 3 To select a background color, click the triangle in the Background Color control and select a color from the palette.
- 4 For Frame Rate, enter the number of animation frames to play each second.
- 5 For publish settings, click the Edit button next to the Profile property. Flash displays the Publish Settings dialog box with the Flash tab selected. Choose appropriate options from the dialog box. For more information, see “[Publishing Flash documents](#)” on page 379.

Add XMP metadata to a document

You can include Extensible Metadata Platform (XMP) data such as title, author, description, copyright, and more in your FLA files. XMP is a metadata format that certain other Adobe applications can understand. The metadata is viewable in Flash and in Adobe® Bridge. For more information about XMP metadata, see *Metadata and Keywords* in Bridge Help.

Embedding metadata improves the ability of web-based search engines to return meaningful search results for Flash content. The search metadata is based on the XMP (Extensible Metadata Platform) specifications and is stored in the FLA file in a W3C-compliant format.

A file’s metadata contains information about the contents, copyright status, origin, and history of the file. In the File Info dialog box, you can view and edit the metadata for the current file.

Depending on the selected file, the following types of metadata may appear:

Description Contains author, title, copyright, and other information.

IPTC Displays editable metadata. You can add captions to your files, as well as copyright information. IPTC Core is a specification that was approved by the IPTC (International Press Telecommunications Council) in October 2004. It differs from the older IPTC (IIM, legacy) in that new properties were added, some property names were changed, and some properties were deleted.

Camera Data (Exif) Displays information assigned by digital cameras, including the camera settings used when the image was taken.

Video Data Displays metadata for video files, including pixel aspect ratio, scene, and shot.

Audio Data Displays metadata for audio files, including artist, album, track number, and genre.

Mobile SWF Lists information about SWF files, including title, author, description, and copyright.

History Keeps a log of changes made to images with Photoshop.

Note: The History Log preference must be turned on in Photoshop for the log to be saved with the file’s metadata.

Version Cue Lists any Version Cue file-version information.

DICOM Displays information about images saved in the Digital Imaging and Communications in Medicine (DICOM) format.

To add metadata:

- 1 Choose File > File Info.
- 2 In the File Info dialog box that appears, enter the metadata that you want to include. You can add or remove metadata in the FLA file at any time.

View a document when multiple documents are open

When you open multiple documents, tabs at the top of the Document window identify the open documents and let you easily navigate among them. Tabs appear only when documents are maximized in the Document window.

- ❖ Click the tab of the document you want to view.

By default, tabs appear in the order in which the documents were created. You can drag the document tabs to change their order.

Save Flash documents

You can save a Flash FLA document using its current name and location or using a different name or location.

When a document contains unsaved changes, an asterisk (*) appears after the document name in the document title bar, the application title bar, and the document tab. When you save the document, the asterisk is removed.

Save a Flash document

- 1 Do one of the following:
 - To overwrite the current version on the disk, select File > Save.
 - To save the document in a different location and/or with a different name, or to compress the document, select File > Save As.
- 2 If you selected Save As, or if the document has never been saved before, enter the filename and location.
- 3 Click Save.

Revert to the last saved version of a document

- ❖ Select File > Revert.

Save a document as a template

- 1 Select File > Save As Template.
- 2 In the Save As Template dialog box, enter a name for the template in the Name box.
- 3 Select a category from the Category pop-up menu, or enter a name to create a new category.
- 4 Enter a description of the template in the Description box (up to 255 characters), and click OK.

The description appears when the template is selected in the New Document dialog box.

Save a document as a Flash CS3 document

- 1 Select File > Save As.
- 2 Enter the filename and location.

3 Select Flash CS3 Document from the Format pop-up menu, and click Save.

Important: *If an alert message indicates that content will be deleted if you save in Flash CS3 format, click Save As Flash CS3 to continue. This might happen if your document contains features that are available only in Flash CS4. Flash does not preserve these features when you save the document in Flash CS3 format.*

Save documents when quitting Flash

- 1 Select File > Exit (Windows) or Flash > Quit Flash (Macintosh).
- 2 If you have documents open with unsaved changes, Flash prompts you to save or discard the changes for each document.
 - Click Yes to save the changes and close the document.
 - Click No to close the document without saving the changes.

Working with other Adobe applications

Flash is designed to work with other Adobe® applications to enable a broad range of creative workflows. You can import Illustrator® and Photoshop® files directly into Flash. You can also create video from Flash and edit it in Adobe® Premiere® Pro or After Effects®, or import video from either of those applications into Flash. When publishing your SWF files, you can use Dreamweaver® to embed the content in your web pages and launch Flash directly from within Dreamweaver to edit the content.

See also

[“Working with Illustrator and Flash”](#) on page 70

[“Working with Photoshop and Flash”](#) on page 78

[“Working with Adobe Premiere Pro and After Effects”](#) on page 294

[“Edit a SWF file from Dreamweaver in Flash”](#) on page 48

Opening XFL files

An XFL file is a type of Flash file that stores the same information as a FLA file, but in XML format. XFL is a set of XML files and other assets (JPEG, GIF, FLV, MP3, WAV, etc.) saved into a compressed XFL package file.

Other Adobe applications, such as InDesign® and After Effects can export files in XFL format. This allows you to work on a project in a separate application first and then continue working with it in Flash.

You can open and work with XFL files in Flash in the same way you would open an FLA file. When you open an XFL file in Flash, you can then save the file as a FLA file. You cannot save XFL files from within Flash.

To open an XFL file in Flash:

- 1 In another Adobe application, such as InDesign or After Effects, export your work as an XFL file.

The application preserves all of the layers and objects of the original file in the XFL file.

- 2 In Flash, choose File > Open and navigate to the XFL file. Click Open.

The XFL file opens in Flash in the same way as an FLA file. All of the layers of the original file appear in the Timeline and the original objects appear in the Library panel.

You can now work with the file normally.

- 3 To save the file, choose File > Save.

Flash prompts you to name the new FLA file in the Save As dialog box.

- 4 Type a name and save the FLA file.

Note: For a video tutorial about exporting XFL files from After Effects, see *Importing and exporting XFL files between Flash and After Effects* at www.adobe.com/go/lrvid4098_xp.

Edit a SWF file from Dreamweaver in Flash

If you have both Flash and Dreamweaver installed, you can select a SWF file in a Dreamweaver document and use Flash to edit it. Flash does not edit the SWF file directly; it edits the source document (FLA file) and re-exports the SWF file.

- 1 In Dreamweaver, open the Property inspector (Window > Properties).
- 2 In the Dreamweaver document, do one of the following:
 - Click the SWF file placeholder to select it; then in the Property inspector click Edit.
 - Right-click (Windows) or Control-click (Macintosh) the placeholder for the SWF file, and select Edit With Flash from the context menu.

Dreamweaver switches the focus to Flash, and Flash attempts to locate the Flash authoring file (FLA) for the selected SWF file. If Flash cannot locate the Flash authoring file, you are prompted to locate it.

Note: If the FLA file or SWF file is locked, check out the file in Dreamweaver.

- 3 In Flash, edit the FLA file. The Flash Document window indicates that you are modifying the file from within Dreamweaver.
- 4 When you finish making edits, click Done.

Flash updates the FLA file, re-exports it as a SWF file, closes, and then returns the focus to the Dreamweaver document.

Note: To update the SWF file and keep Flash open, in Flash select File > Update for Dreamweaver.

- 5 To view the updated file in the document, click Play in the Dreamweaver Property inspector or press F12 to preview your page in a browser window.

Create mobile content with Adobe Device Central and Flash

- 1 Start Flash.
- 2 On the main Flash screen, select Create New > Flash File (Mobile).

Flash opens Adobe® Device Central and displays the New Document tab.

- 3 In Device Central, select a Player version and ActionScript version.

The Available Devices list on the left is updated. Devices that do not support the selected Player version and ActionScript version are dimmed.

- 4 Select a content type.

The Available Devices list on the left is updated and shows the devices that support the content type (as well as the Player version and ActionScript version) selected.

- 5 In the Available Devices list, select a single target device or multiple devices (or select a set or individual device in the Device Sets list).

Device Central lists proposed document sizes based on the device or devices you selected (if the devices have different display sizes). Depending on the design or content you are developing, you can create a separate mobile document for each display size or try to find one size appropriate for all devices. When choosing the second approach, you may want to use the smallest or largest suggested document size as a common denominator. You can even specify a custom size at the bottom of the tab.

6 Click Create.

Flash starts up and creates a document with preset publish settings from Device Central, including the correct size for the device (or group of devices) specified.

7 Add content to the new Flash document.

8 To test the document, select Control > Test Movie.

The new document is displayed in the Device Central Emulator tab. If one or more devices were selected in the Available Devices list in step 5, a new device set is created (named according to the FLA file) and listed in the Device Sets panel. The device shown in the Emulator tab is listed in the Device Sets panel with a special icon 🍌. To test the new Flash document on another device, double-click the name of a different device in the Device Sets or Available Devices lists.

See also

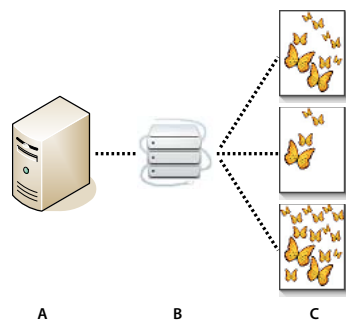
[Using Device Central with Flash](#)

[Creating mobile content in Flash](#)

About Version Cue

Version Cue acts as a virtual server that lets designers work collaboratively on a set of common files. Designers can easily track and manipulate multiple versions of files.

Version Cue consists of two pieces: the Version Cue Server and Adobe Drive. The Version Cue Server can be installed locally or on a dedicated computer, and it hosts Version Cue projects and PDF reviews. Adobe Drive connects to Version Cue CS4 servers. The connected server appears like a hard drive or mapped network drive in Explorer, Finder, and dialog boxes such as Open and Save As.



Version Cue

A. Version Cue server B. Users connect to server using Adobe Drive C. Users check out and save different versions of a file

Use Version Cue to track versions of a file as you work and to enable workgroup collaboration such as file sharing, version control, backups, online reviews, and the ability to check files in and out. You can organize Version Cue-managed files into private or shared projects.

Version Cue is integrated with Adobe Bridge: Use Adobe Bridge as a file browser for Version Cue projects. With Adobe Bridge, you can access Version Cue Servers, projects, and files, and view, search for, and compare information about Version Cue-managed assets.

Use Version Cue Server Administration to create and manage user access, projects, and PDF reviews; administer backups; export content; and specify advanced Version Cue Server information.

Note: Version Cue is not available in Adobe Bridge with Adobe Photoshop® Elements for Macintosh.

For a video on setting up Version Cue, see www.adobe.com/go/lrvid4038_vc.

See also

[Version Cue video](#)

Printing Flash documents

Print from Flash documents

To preview and edit your documents, print frames from Adobe® Flash® CS4 Professional documents, or specify frames to be printable from Flash Player by a viewer.

When printing frames from a Flash document, use the Print dialog box to specify the range of scenes or frames to print and the number of copies. In Windows, the Page Setup dialog box specifies paper size, orientation, and various print options—including margin settings and whether all frames are to be printed for each page. On the Macintosh, these options are divided between the Page Setup and the Print Margins dialog boxes.

The Print and Page Setup dialog boxes are standard in either operating system, and their appearance depends on the selected printer driver.

- 1 Select File > Page Setup (Windows) or File > Print Margins (Macintosh).
- 2 Set page margins. Select both Center options to print the frame in the center of the page.
- 3 In the Frames menu, select whether to print all frames in the document or only the first frame of each scene.
- 4 In the Layout menu, select from the following options:

Actual Size Prints the frame at full size. Enter a value for Scale to reduce or enlarge the printed frame.

Fit On One Page Reduces or enlarges each frame so it fills the print area of the page.

Storyboard Prints several thumbnails on one page. Select from Boxes, Grid, or Blank. Enter the number of thumbnails per page in the Frames box. Set the space between the thumbnails in the Frame Margin box, and select Label Frames to print the frame label as a thumbnail.

- 5 To print frames, select File > Print.

Use frame labels to disable printing

To choose not to print any of the frames in the main Timeline, label a frame as !#p to make the entire SWF file nonprintable. Labeling a frame as !#p dims the Print command in the Flash Player context menu. You can also remove the Flash Player context menu.

If you disable printing from Flash Player, the user can still use the browser Print command to print frames. Because this command is a browser feature, you cannot use Flash to control or disable it.

See also

[“Publishing overview”](#) on page 379

Disable printing in the Flash Player context menu

- 1 Open or make active the Flash document (FLA file) to publish.
- 2 Select the first keyframe in the main Timeline.
- 3 Select Window > Properties to view the Property inspector.
- 4 In the Property inspector, for Frame Label enter !#p to specify the frame as non-printing.

Specify only one !#p label to dim the Print command in the context menu.

Note: You can also select a blank frame (rather than a keyframe) and label it #p.

Disable printing by removing the Flash Player context menu

- 1 Open or make active the Flash document (FLA file) to publish.
- 2 Select File > Publish Settings.
- 3 Select the HTML tab and deselect Display Menu and click OK.

Specify a print area when printing frames

- 1 Open the Flash document (FLA file) containing the frames you will set to print.
- 2 Select a frame that you have not specified to print with a #p frame label that is on the same layer as a frame that is labeled with a #p.

To organize your work, select the next frame after a frame labeled #p.

- 3 Create a shape on the Stage the size of the desired print area. To use a frame’s bounding box, select a frame with any object of the appropriate print area size.
- 4 Select the frame in the Timeline that contains the shape to use for the bounding box.
- 5 In the Property inspector (Window > Properties), enter #b for Frame Label to specify the selected shape as the bounding box for the print area.

Only one #b frame label per Timeline is allowed. This option is the same as selecting the Movie bounding box option with the Print action.

Change the printed background color

You can print the background color set in the Document Properties dialog box. Change the background color for only the frames to be printed by placing a colored object on the lowest layer of the Timeline being printed.

- 1 Place a filled shape that covers the Stage on the lowest layer of the Timeline that will print.
- 2 Select the shape and select Modify > Document. Select a color for the printing background.

This action changes the entire document’s background color, including that of movie clips and loaded SWF files.

- 3 Do one of the following:
 - To print that color as the document’s background, designate to print the frame in which you placed the shape.
 - To maintain a different background color for non-printing frames, repeat steps 2 and 3. Then place the shape on the lowest layer of the Timeline, in all the frames that are not designated to print.

Print from the Flash Player context menu

Use the Print command in the Flash Player context menu to print frames from any Flash SWF file.

The context menu's Print command cannot print transparency or color effects and cannot print frames from other movie clips; for more advanced printing capabilities, use the `PrintJob` object or the `print()` function.

1 Open the document.

The command prints the frames labeled #p by using the Stage for the print area or the specified bounding box.

If you haven't designated specific frames to print, all frames in the document main Timeline print.

2 Select File > Publish Preview > Default or press F12 to view your Flash content in a browser.

3 Right-click (Windows) or Control-click (Macintosh) in the Flash content in the browser window to display the Flash Player context menu.

4 Select Print from the Flash Player context menu to display the Print dialog box.

5 In Windows, select the print range to select which frames to print.

6 On the Macintosh, in the Print dialog box, select the pages to print.

7 Select other print options, according to your printer's properties.

8 Click OK (Windows) or Print (Macintosh).

Note: Printing from the context menu does not interact with calls to the `PrintJob` object.

Working with projects

About projects

You can use Flash projects to manage multiple document files in a single project. Flash projects allow you to group multiple, related files together to create complex applications.

A Flash project can contain any Flash or other file type, including previous versions of FLA and SWF files.

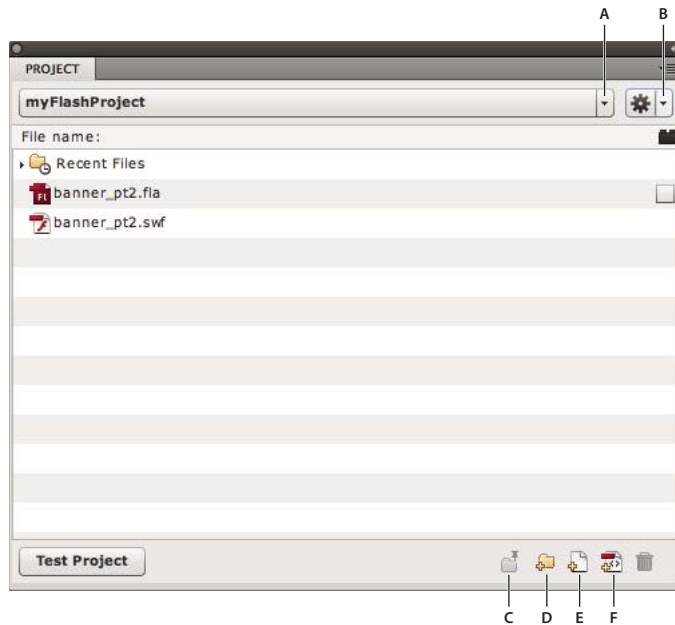
Create and manage projects

You use the Project panel (Window > Other Panels > Project) to create and manage projects. The panel displays the contents of a Flash project in a collapsible tree structure. The panel title bar displays the project name.

If certain types of project files are missing (not in the specified location), a dialog specifying which items are missing will alert you to the missing files, and give you an opportunity to browse for their locations. Missing files are most commonly external folders not inside the root project folder. All other files are automatically updated based on the root project folder location.

When you publish a project, each FLA file in the Publish List (denoted by a check mark next to the FLA name) is published with the publish profile specified for that file.

Only one project can be open at one time. If a project is open and you open or create another project, Flash automatically saves and closes the first project.



The Project panel
 A. Project menu B. Options menu C. Pin D. New folder E. New file F. Create class

See also

“[Using publish profiles](#)” on page 397

Create a project

- 1 From the Project menu select New Project.
- 2 Browse for a folder.
- 3 Choose a name for your project. The Project panel uses the name of the folder containing the project by default.
- 4 From the ActionScript Version menu, choose the type of project you would like to create: ActionScript 3.0 or ActionScript 2.0.

Flash adds the project to the Project menu, and the Project Panel displays the contents of the directory that contains the selected folder.

Flash reads the directory of the initial folder you choose. To change directories, create another project.

Create a Quick Project

The Project panel allows you to quickly create a project based on the currently open FLA file. A Quick Project is identical to any other project.

- ❖ Select Quick Project from the Project menu.

Flash creates a project using the front most FLA file as the default document. The Quick Project will be named after the FLA file.

To remove the Quick Project, delete it from the Project menu.

Multiple Quick Projects are allowed.

***Note:** Creating a Quick Project from a FLA file that resides in the root of another project changes the name of the other project to the name of the FLA file, and sets the default document to the FLA file.*

Open an existing project

- 1 Select Window > Other Panels > Project to open the Project panel.
- 2 Select Open Project from the Project pop-up menu in the Project panel.
- 3 Using the Browse For Folder dialog box, navigate into the folder containing the project and click OK.

Note: You cannot open Dreamweaver projects in the Flash Project panel.

Close a project

- ❖ Select Close Project from the Options pop-up menu.

Delete a project

- 1 Open the Project you want to delete.
- 2 From the Options pop-up menu select Delete Project.
- 3 Select how to delete the project:
 - Delete the currently active project from the Project Panel.
 - Delete the currently active project from the Project Panel and all of the files stored in the project directory. This deletes all of the selected files from the file system.

Flash removes the selected project from the Project panel.

Convert previous projects to the new project format

In previous versions of Flash, projects used an XML file with the filename extension .flp—for example, myProject.flp. To use projects created in previous versions of Flash, you must convert the old project to the new format.

- 1 Select Window > Other Panels > Project to open the Project panel.
- 2 Select Open Project from the Project pop-up menu in the Project panel.
- 3 Using the Browse For Folder dialog box, navigate to the folder containing the FLP file, and click OK, specifying that folder as the converted project's root folder.

The contents of the specified folder will be displayed in the Project panel.

Add a new file or folder to a project

The Project panel allows you to open, create, and remove files and folders.

The Project panel filters files by filename extension. However, you can manually hide files by preceding the filename with a special character, such as an underscore. To enable this feature, enter the special characters to filter in the “Hide files or folders beginning with” section of the Panel Preferences (Options > Panel Preferences).

By default, the Project panel only shows Flash document types (FLA, SWF, SWC, AS, JSFL, ASC, MXML, TXT, XML), but you can add custom types in the Settings tab of the Panel Preferences (Options > Panel Preferences).

The Project panel shows all files opened recently in the panel in the Recent Files folder at the top of the project tree.

Recent Files are sorted in the order they were opened. Only files opened with the Project panel appear in this list. To disable/enable the recent files folder, toggle the Show Recent Files checkbox in the Settings Tab of the Panel Preferences. You can also specify how many files are shown in the list.

To clear the Recent Files list, select Clear Recent Files from the Options menu, or right-click the Recent Files folder, and select Clear Recent Files from the context menu. Clearing the recent files list does not remove the files from your computer.

- 1 Select the folder in which you want to create the file or folder. If there is no selection, Flash creates the file or folder in the project root folder.
- 2 Click the New File or New Folder button at the bottom of the panel.
- 3 Choose a name and a file type (files only). Select the Open File After Creation checkbox to open the new file in Flash. Then click Create File.

Note: If a file or folder with the name you've specified already exists, a dialog alerts you about the existing file or folder.

The new file appears in the file list.

Rename a project

- 1 Select the project name in the Project panel.
- 2 Select Rename Project from the Options pop-up menu.
- 3 Enter a new name and click OK.

Open a file from the Project panel in Flash

- ❖ Double-click the filename in the Project panel.

If the file is of a native file type (a type supported by the Flash authoring tool), the file opens in Flash. If the file is a non-native file type, the file opens in the application used to create it.

Remove a file or folder from a project

- 1 Select the file or folder you want to remove from the file list.
- 2 Click the Delete icon in the far right of the icon tray, or press the Delete key.

You can also right-click on the file or folder and choose Delete from the context menu.

- 3 In the dialog box that appears, confirm your selection.

Flash deletes the file from your hard drive completely. Deleting a file or directory cannot be undone. Deleting a folder also removes its files and subdirectories.

Add an external folder to a project

The Project panel allows for the addition of folders external to the project root folder, referred to as *locations*. Folder paths can be added (and shown or hidden) on a global basis, or per project. Locations act as any other folder, and have all the same functionality as folders in the project root folder.

- 1 Open the Panel or Project Preferences.
- 2 Select the Locations tab.
- 3 To make locations visible in the project file list, select the “Specify additional source folders outside of the root project folder” checkbox.
- 4 Click the “+” button.
- 5 Browse to the external folder you want to add.
- 6 Give the location a name.

Location names can be changed at any time in the Project or Panel preferences by selecting the desired location and clicking on the Rename Location button.

- 7 When you are finished adding locations, click OK to close the dialog. Location folders appear above all other folders in the file list.

You cannot delete or remove a location in the operating system file browser. Remove them using the Locations tab in the Project or Panel preferences.

Create ActionScript classes

The Project panel supports creating template-based ActionScript classes within an ActionScript package structure. Use the Create Class button to add class files to specified packages. ActionScript 3.0 and ActionScript 2.0 templates are supported.

Use the panel preferences to specify options for classes. If you specify a folder in the Save Classes In text box in the Classes tab of the Panel preferences, the Project panel shows that folder in the project directory with a <> symbol to indicate that it is the default location for new classes. Flash adds this folder to the class path when publishing via the Project panel.

This class path folder can reside in a subdirectory of your project folder, in a directory relative to your project folder, or anywhere on your computer. You can set the class path globally in the Panel preferences, or per project in the Project preferences. By default, all class packages are stored in the root of your project.

The classes' folder visibility can be toggled in the Project panel preferences Settings tab.

- 1 Click the Create Class button at the bottom of the Project panel.
- 2 Enter the package path of your class.
- 3 Select Bind Class to Library Symbol to create a movie clip in your library (in the same folder structure as your package path). To bind to an existing movie clip, select a movie clip in the library, and select Use Selected Library Symbol in the Create Class dialog box. Flash moves the movie clip to the appropriate location in the library, and updates the symbol linkage to reflect the new class.
- 4 Select Declare Stage Instances to add variable declarations and import statements to the new class file to account for all of the named instances on the timeline of the selected movie clip.
- 5 Select Open File After Creation to open your class file for editing after it is created.
- 6 Click Create Class.

The Project panel creates the class file based on the class template, and places it in the appropriate package directory, creating directories if necessary.

Clicking the Create Class button does not overwrite class files. If the file exists, it does not create a new file. This can be useful if you would like to bind an existing class to a movie clip, move the movie clip into the appropriate library folder, and set the appropriate linkages.

Create class templates

The Project panel supports a simple template structure, used when creating class files. This system is composed of four files. All the base files are located in the Flash Configuration folder, in the /Project/templates/ subdirectory.

The four files are:

- 1 boundClass_as2.as
- 2 boundClass_as3.as
- 3 standardClass_as2.as

4 standardClass_as3.as

Each of these template files has custom template tags that are replaced with dynamic data when a new class is created. You can place these tags anywhere in your class template file.

The following are the tags included in each file:

- All classes:
 - %PACKAGE_NAME%
 - %CLASS_NAME%
 - %BASE_CLASS_NAME% (AS3 only)
- Bound class only:
 - %LINKAGE_ID%
 - %AUTO_IMPORTS%
 - %AUTO_ELEMENTS%

To create a template file, you can either modify the template files, or create your own class template. To change the default class templates, open the Panel Preferences and browse to the correct template file. Flash then uses this file for all your projects.

The Project preferences dialog contains a similar tab that allows you to set template files on a per-project basis.

The templates set in the Panel preferences is only used when creating new projects. To change a template path for a project, set it in the Project preferences. Template paths in the Project and Panel preferences must be absolute. Relative paths are not supported.

Pin a directory

When dealing with large projects, you can reduce clutter by only showing files in a specific directory.

- 1 Select the directory you would like to pin.
- 2 Click the Pin Directory button.

The Pin Directory button changes to the Unpin Directory button, and the selected directory appears at the root of the project listing.

To pin a different directory, un-pin the current pinned directory first.

Locations and the classes folder (and their subdirectories) can also be pinned. When a location or the classes folder (or subfolder) is pinned, the classes folder is not visible.

Compiling FLA files

- To test, publish or export a FLA file, right-click (Windows) or command-click (Macintosh) on any FLA file in the project tree and select the appropriate action from the context menu.
- To test a project, click the Test Project button.

Flash compiles the movie specified as the default document and opens the resulting SWF file. You can change the default document at any time by right-clicking (Windows) or Command-clicking (Macintosh) on any FLA file and choosing Make Default Document from the context menu.

The Project panel allows you to create a publish list. This is a list of FLA files that can be batch-compiled at any time.

- To add or remove files from the publish list, select the checkbox to the right of each FLA file in the project list.

- To publish the files in the publish list, click Publish List in the Project panel Options menu.

If one of the FLA files has errors when compiling, the Project panel stops compiling the list and displays the error in the Output panel.

Create Flash projects for use with Flex class files

Flash lets you create skins and other visual elements for use in Flex, as well as allowing you to create ActionScript code in Flex but publish the resulting SWF file using Flash. To support the use Flex class files in Flash, the Project panel provides support for identifying the Flex SDK.

To use Flex classes in Flash, you must specify the location of the Flex SDK.

- 1 Create a project.
- 2 Select Project Properties from the Options pop-up menu.
- 3 Select the Paths tab, and enter the directory path or browse to the Flex SDK. For example: C:\Program Files\Adobe FlexBuilder 3\sdk\3.0.0
- 4 From the Paths tab, select the Source tab and add the path to any folders containing ActionScript class files.

Click the Plus (+) button to enter a path, or the Browse button to select a folder containing class files. To delete a path, select it and click the Minus (-) button.

- 5 Select the Library tab, and enter the path to SWC (Flash component) files or folders containing SWC files.

Click the Plus (+) button to enter a path, or the Browse button to select a folder containing SWC files. To delete an SWC file or folder, select it and click the Minus (-) button.

- 6 Select the External Library tab, and enter the path to SWC (Flash component) files or folders containing SWC files loaded as runtime shared libraries.

Click the Plus (+) button to enter a path or the Browse button to select a folder containing SWC files. To delete an SWC file or folder, select it and click the Minus (-) button.

- 7 Click OK.

Find and Replace

About Find and Replace

The Find and Replace feature lets you do the following:

- Search for a text string, a font, a color, a symbol, a sound file, a video file, or an imported bitmap file.
- Replace the specified element with another element of the same type. Different options are available in the Find and Replace dialog box depending on the type of specified element.
- Find and replace elements in the current document or the current scene.
- Search for the next occurrence or all occurrences of an element, and replace the current occurrence or all occurrences.

Note: In a screen-based document, you can find and replace elements in the current document or the current screen, but you can't use scenes.

The Live Edit option lets you edit the specified element directly on the Stage. If you use Live Edit when searching for a symbol, Flash opens the symbol in edit-in-place mode.

The Find and Replace Log at the bottom of the Find and Replace dialog box shows the location, name, and type of the elements for which you are searching.

See also

“[Working with screens](#)” on page 316

Find and replace text

- 1 Select Edit > Find and Replace.
- 2 Select Text from the For pop-up menu.
- 3 In the Text box, enter the text to find.
- 4 In the Replace With Text box, enter the text to replace the existing text.
- 5 Select options for searching text:

Whole Word Searches for the specified text string as a whole word only, bounded on both sides by spaces, quotes, or similar markers. When Whole Word is deselected, the specified text can be searched as part of a larger word. For example, when Whole Word is deselected, a search for *place* will yield the words *replace*, *placement*, and so on.

Match Case Searches for text that exactly matches the case (uppercase or lowercase character formatting) of the specified text when finding and replacing.

Regular Expressions Searches for text in regular expressions in ActionScript. An expression is any statement that Flash can evaluate that returns a value.

Text Field Contents Searches the contents of a text field.

Frames/Layers/Parameters Searches frame labels, layer names, scene names, and component parameters.

Strings in ActionScript Searches strings (text between quotation marks) in ActionScript in the document or scene (external ActionScript files are not searched).

ActionScript Searches all ActionScript, including code and strings.

- 6 To select the next occurrence of the specified text on the Stage and edit it in place, select Live Edit.

Note: Only the next occurrence is selected for live editing, even if you select Find All in step 7.

- 7 To find text, do one of the following:
 - To find the next occurrence of the specified text, click Find Next.
 - To find all occurrences of the specified text, click Find All.
- 8 To replace text, do one of the following:
 - To replace the currently selected occurrence of the specified text, click Replace.
 - To replace all occurrences of the specified text, click Replace All.

Find and replace fonts

- 1 Select Edit > Find and Replace.
- 2 Select Font from the For pop-up menu, then select from the following options:
 - To search by font name, select Font Name and select a font from the pop-up menu or enter a font name in the box. When Font Name is deselected, all fonts in the scene or document are searched.

- To search by font style, select Font Style and select a font style from the pop-up menu. When Font Style is deselected, all font styles in the scene or document are searched.
- To search by font size, select Font Size and enter values for minimum and maximum font size to specify the range of font sizes to be searched. When Font Size is deselected, all font sizes in the scene or document are searched.
- To replace the specified font with a different font name, select Font Name under Replace With and select a font name from the pop-up menu or enter a name in the box. When Font Name is deselected under Replace with, the current font name remains unchanged.
- To replace the specified font with a different font style, select Font Style under Replace With and select a font style from the pop-up menu. When Font Style is deselected under Replace with, the current style of the specified font remains unchanged.
- To replace the specified font with a different font size, select Font Size under Replace With and enter values for minimum and maximum font size. When Font Size is deselected under Replace With, the current size of the specified font remains unchanged.

3 To select the next occurrence of the specified font on the Stage and edit it in place, select Live Edit.

***Note:** Only the next occurrence is selected for live editing, even if you select Find All in step 4.*

4 To find a font, do one of the following:

- To find the next occurrence of the specified font, click Find Next.
- To find all occurrences of the specified font, click Find All.

5 To replace a font, do one of the following:

- To replace the currently selected occurrence of the specified font, click Replace.
- To replace all occurrences of the specified font, click Replace All.

Find and replace colors

You cannot find and replace colors in grouped objects.

***Note:** To find and replace colors in a GIF or JPEG file in a Flash document, edit the file in an image-editing application.*

1 Select Edit > Find and Replace.

2 Select Color from the For pop-up menu.

3 To search for a color, click the Color control and do one of the following:

- Select a color swatch from the color pop-up window.
- Enter a hexadecimal color value in the Hex Edit box in the color pop-up window.
- Click the System Color Picker button and select a color from the system color picker.
- To make the eyedropper tool appear, drag from the Color control. Select any color on your screen.

4 To select a color to replace the specified color, click the Color control under Replace With and do one of the following:

- Select a color swatch from the color pop-up window.
- Enter a hexadecimal color value in the Hex Edit box in the color pop-up window.
- Click the System Color Picker button and select a color from the system color picker.
- To make the eyedropper tool appear, drag from the Color control. Select any color on your screen.

5 To specify which occurrence of the color to find and replace, select the Fills, Strokes, or Text option or any combination of those options.

6 To select the next occurrence of the specified color on the Stage and edit it in place, select Live Edit.

Note: Only the next occurrence is selected for live editing, even if you select Find All in the next step.

7 Find a color.

- To find the next occurrence of the specified color, click Find Next.
- To find all occurrences of the specified color, Click Find All.

8 Replace a color.

- To replace the currently selected occurrence of the specified color, click Replace.
- To replace all occurrences of the specified color, click Replace All.

Find and replace symbols

To find and replace symbols, search for a symbol by name. Replace a symbol with another symbol of any type—movie clip, button, or graphic.

1 Select Edit > Find and Replace.

2 Select Symbol from the For pop-up menu.

3 For Name, select a name from the pop-up menu.

4 Under Replace With, for Name select a name from the pop-up menu.

5 To select the next occurrence of the specified symbol on the Stage and edit it in place, select Live Edit.

Note: Only the next occurrence is selected for editing, even if you select Find All in the next step.

6 To find a symbol, do one of the following:

- To find the next occurrence of the specified symbol, click Find Next.
- To find all occurrences of the specified symbol, click Find All.

7 To replace a symbol, do one of the following:

- To replace the currently selected occurrence of the specified symbol, click Replace.
- To replace all occurrences of the specified symbol, click Replace All.

Find and replace sound, video, or bitmap files

1 Select Edit > Find and Replace.

2 Select Sound, Video, or Bitmap from the For pop-up menu.

3 For Name, enter a sound, video, or bitmap filename or select a name from the pop-up menu.

4 Under Replace With, for Name enter a sound, video, or bitmap filename or select a name from the pop-up menu.

5 To select the next occurrence of the specified sound, video, or bitmap on the Stage and edit it in place, select Live Edit.

Note: Only the next occurrence is selected for editing, even if you select Find All in the next step.

6 Find a sound, video, or bitmap.

- To find the next occurrence of the specified sound, video, or bitmap, click Find Next.

- To find all occurrences of the specified sound, video, or bitmap, click Find All.
- 7 Replace a sound, video, or bitmap.
- To replace the currently selected occurrence of the specified sound, video, or bitmap, click Replace.
 - To replace all occurrences of the specified sound, video, or bitmap, click Replace All.

Templates

Templates are FLA files that are preconfigured with content that you can customize for your own purposes. There are several different templates available.

About templates

Flash templates provide you with easy-to-use starting points for common projects. There are several templates for advertising projects included with Flash.

For more information on authoring Flash files for mobile devices, see the Adobe Mobile Devices site at www.adobe.com/go/devnet_devices.

Advertising templates

Advertising templates facilitate the creation of standard rich media types and sizes defined by the Interactive Advertising Bureau (IAB) and accepted by the industry. For more information on IAB-endorsed ad types, see the IAB site at IAB.net.

Test ads for stability in a variety of browser and platform combinations. Your application is considered stable if it doesn't cause error messages, browser crashes, or system crashes.

Work with web masters and network administrators to create detailed testing plans that include tasks relevant to your users. Make these plans publicly available and update them regularly. Vendors should publish detailed plans indicating the browser and platform combinations in which their technologies are stable. Examples are available at the IAB Rich Media testing section of IAB.net. Size and file format requirements of ads might vary by vendor and site. Check with your vendor, ISP, or the IAB to learn about these requirements that can affect the ad's design.

Use templates

- 1 Select File > New.
- 2 Click the Templates tab.
- 3 Select a template and click OK.
- 4 Add content to the FLA file.
- 5 Save and publish the file.

Chapter 4: Using imported artwork

A key advantage of Adobe® Flash® CS4 Professional is that you can import artwork created in other applications and use those assets in your Flash documents. You can import vector graphics and bitmap images in a variety of file formats, making Adobe® Flash® CS4 Professional a versatile media arts tool.

When you import a bitmap, you can apply compression and anti-aliasing, place the bitmap directly in an Adobe® Flash® CS4 Professional document, use the bitmap as a fill, edit the bitmap in an external editor, break the bitmap apart into pixels and edit it in Adobe® Flash® CS4 Professional, or convert the bitmap to vector artwork.

Placing artwork into Flash

About importing artwork into Flash

Adobe® Flash® CS4 Professional can use artwork created in other applications. You can import vector graphics and bitmaps in a variety of file formats. If you have QuickTime® 4 or later installed on your system, you can import additional vector or bitmap file formats. You can import Adobe® FreeHand® files (version MX and earlier) and Adobe® Fireworks® PNG files directly into Flash, preserving attributes from those formats.

Graphic files that you import into Flash must be at least 2 pixels x 2 pixels in size.

To load JPEG files into a Flash SWF file during runtime, use the `loadMovie` action or method. For detailed information, see `loadMovie` (`MovieClip.loadMovie` method) in [ActionScript 2.0 Language Reference](#) or Working with movie clips in *Programming ActionScript 3.0*.

Flash imports vector graphics, bitmaps, and sequences of images as follows:

- When you import Adobe® Illustrator® and Adobe® Photoshop® files into Flash, you can specify import options that let you preserve most of your artwork's visual data, and the ability to maintain the editability of certain visual attributes with the Flash authoring environment.
- When you import vector images into Flash from FreeHand, select options for preserving FreeHand layers, pages, and text blocks.
- When you import PNG images from Fireworks, import files as editable objects to modify in Flash, or as flattened files to edit and update in Fireworks.
- Select options for preserving images, text, and guides.

Note: *If you import a PNG file from Fireworks by cutting and pasting, the file is converted to a bitmap.*

- Vector images from SWF and Windows® Metafile Format (WMF) files that you import directly into a Flash document (instead of into a library) are imported as a group in the current layer.
- Bitmaps (scanned photographs, BMP files) that you import directly into a Flash document are imported as single objects in the current layer. Flash preserves the transparency settings of imported bitmaps. Because importing a bitmap can increase the file size of a SWF file, consider compressing imported bitmaps.

Note: *Bitmap transparency might not be preserved when bitmaps are imported by dragging and dropping from an application or desktop to Flash. To preserve transparency, use the File > Import To Stage or Import To Library command for importing.*

- Any sequence of images (for example, a PICT or BMP sequence) that you import directly into a Flash document is imported as successive keyframes of the current layer.

See also

[“Imported bitmaps”](#) on page 86

[“Video”](#) on page 283

[“Sound”](#) on page 273

[“Set bitmap properties”](#) on page 87

Import artwork in Flash

Flash lets you import artwork in various file formats either directly to the Stage, or to the library.

See also

[“Symbols, instances, and library assets”](#) on page 147

Import a file into Flash

1 Do one of the following:

- To import a file directly into the current Flash document, select File > Import > Import To Stage.
- To import a file into the library for the current Flash document, select File > Import > Import To Library. (To use a library item in a document, drag it onto the Stage.)

2 Select a file format from the Files Of Type (Windows) or Show (Macintosh) pop-up menu.

3 Navigate to the desired file and select it. If an imported file has multiple layers, Flash might create new layers (depending on the import file type). Any new layers appear in the Timeline.

4 Click Open.

5 If the name of the file you are importing ends with a number and additional sequentially numbered files are in the same folder, do one of the following:

- To import all the sequential files, click Yes.
- To import only the specified file, click No.

The following are examples of filenames that can be used as a sequence:

Frame001.gif, Frame002.gif, Frame003.gif

Bird 1, Bird 2, Bird 3

Walk-001.ai, Walk-002.ai, Walk-003.ai

Paste a bitmap from another application directly into the current Flash document

- 1 Copy the image in the other application.
- 2 In Flash, select Edit > Paste In Center.

Supported file formats for vector or bitmap files

Flash can import different vector or bitmap file formats depending on whether QuickTime 4 or later is installed on your system. Using Flash with QuickTime 4 installed is especially useful for collaborative projects in which authors work on both Windows and Macintosh platforms. QuickTime 4 extends support for certain file formats (including PICT, QuickTime Movie, and others) to both platforms.

You can import the following vector or bitmap file formats into Flash 8 or later, regardless of whether QuickTime 4 is installed:

File type	Extension	Windows	Macintosh
Adobe Illustrator (version 10 or earlier)	.ai	•	•
Adobe Photoshop	.psd	•	•
AutoCAD® DXF	.dxf	•	•
Bitmap	.bmp	•	•
Enhanced Windows Metafile	.emf	•	
FreeHand	.fh7, .fh8, .fh9, .fh10, .fh11	•	•
FutureSplash Player	.spl	•	•
GIF and animated GIF	.gif	•	•
JPEG	.jpg	•	•
PNG	.png	•	•
Flash Player 6/7	.swf	•	•
Windows Metafile	.wmf	•	•

You can import the following bitmap file formats into Flash only if QuickTime 4 or later is installed:

File type	Extension	Windows	Macintosh
MacPaint	.pntg	•	•
PICT	.pct, .pic	• (As bitmap)	•
QuickTime Image	.qtif	•	•
Silicon Graphics Image	.sgi	•	•
TGA	.tga	•	•
TIFF	.tif	•	•

Import FreeHand MX files

You can import FreeHand files in version 7 or later directly into Flash. FreeHand is a good choice for creating vector graphics for import into Flash, because you can preserve FreeHand layers, text blocks, library symbols, and pages, and choose a page range to import. If the imported FreeHand file is in CMYK color mode, Flash converts the file to RGB.

Keep the following guidelines in mind when importing FreeHand files:

- When importing a file with overlapping objects to preserve as separate objects, place the objects on separate layers in FreeHand, and select Layers in the FreeHand Import dialog box in Flash. (If overlapping objects on a single layer are imported into Flash, the overlapping shapes are divided at intersection points, just as with overlapping objects that you create in Flash.)
- Flash can support up to eight colors in a gradient fill. If a FreeHand file contains a gradient fill with more than eight colors, Flash creates clipping paths to simulate the appearance of a gradient fill. Clipping paths can increase file size. To minimize file size, use gradient fills with eight colors or fewer in FreeHand.
- Flash imports each step in a blend as a separate path. Thus, the more steps a blend has in a FreeHand file, the larger the imported file size is in Flash.
- When you import files with strokes that have square caps, Flash converts the caps to round caps.
- Flash converts placed grayscale images to RGB images. This conversion can increase the imported file's size.
- When you import files with placed EPS images, select the Convert Editable EPS When Imported option in FreeHand Import Preferences before you place the EPS into FreeHand. If you do not select this option, the EPS image is not viewable when imported into Flash. In addition, Flash does not display information for an imported EPS image (regardless of the Preferences settings used in FreeHand).

1 Select File > Import > Import To Stage or File > Import To Library.

2 Select FreeHand from the Files Of Type (Windows) or Show (Macintosh) pop-up menu.

3 Navigate to a FreeHand file and select it.

4 Click Open.

5 For Mapping Pages, select a setting:

Scenes Converts each page in the FreeHand document to a scene in the Flash document.

Keyframes Converts each page in the FreeHand document to a keyframe in the Flash document.

6 For Mapping Layers, select one of the following:

Layers Converts each layer in the FreeHand document to a layer in the Flash document.

Keyframes Converts each layer in the FreeHand document to a keyframe in the Flash document.

Flatten Converts all layers in the FreeHand document to a single flattened layer in the Flash document.

7 For Pages, do one of the following:

- To import all pages from the FreeHand document, select All.
- To import a page range from the FreeHand document, enter page numbers for From and To.

8 For Options, select any of the following options:

Include Invisible Layers Imports all layers (visible and hidden) from the FreeHand document.

Include Background Layer Imports the background layer with the FreeHand document.

Maintain Text Blocks Preserves text in the FreeHand document as editable text in the Flash document.

9 Click OK.

About AutoCAD DXF files

Flash supports the AutoCAD® DXF format in AutoCAD 10.

DXF files do not support the standard system fonts. Flash tries to map fonts appropriately, but the results can be unpredictable, particularly for text alignment.

Because the DXF format does not support solid fills, filled areas are exported as outlines only. For this reason, the DXF format is most appropriate for line drawings, such as floor plans and maps.

You can import two-dimensional DXF files into Flash. Flash does not support three-dimensional DXF files.

Although Flash doesn't support scaling in a DXF file, all imported DXF files produce 12-inch x 12-inch files that you can scale using Modify > Transform > Scale. Also, Flash supports only ASCII DXF files. If your DXF files are binary, convert them to ASCII before importing them into Flash.

Working with Fireworks files

About imported Fireworks PNG files

You can import Adobe® Fireworks PNG files into Flash as flattened images or as editable objects. When you import a PNG file as a flattened image, the entire file (including any vector artwork) is rasterized, or converted to a bitmap image. When you import a PNG file as editable objects, vector artwork in the file is preserved in vector format. Choose to preserve placed bitmaps, text, filters (called effects in Fireworks), and guides in the PNG file when you import it as editable objects.

About imported filters and blends from Fireworks PNG files

When you import Fireworks® PNG files, you can retain many of the filters and blending modes applied to objects in Fireworks and continue to modify those filters and blends using Flash.

Flash only supports modifiable filters and blends for objects imported as text and movie clips. If an effect or blend mode is not supported, Flash rasterizes or ignores it when it is imported. To import a Fireworks PNG file that contains filters or blends that Flash does not support, rasterize the file during the import process. After this operation, you cannot edit the file.

Fireworks effects supported in Flash

Flash imports the following Fireworks effects as modifiable filters:

Fireworks effect	Flash filter
Drop shadow	Drop shadow
Solid shadow	Drop shadow
Inner shadow	Drop shadow (with Inner shadow automatically selected)
Blur	Blur (where blurX = blurY=1)
Blur more	Blur (where blurX = blurY=1)
Gaussian blur	Blur
Adjust color brightness	Adjust color
Adjust color contrast	Adjust color

Fireworks blend modes supported in Flash

Flash imports the following Fireworks blend modes as modifiable blends:

Fireworks blending mode	Flash blending mode
Normal	Normal
Darken	Darken
Multiply	Multiply
Lighten	Lighten
Screen	Screen
Overlay	Overlay
Hard light	Hard light
Additive	Add
Difference	Difference
Invert	Invert
Alpha	Alpha
Erase	Erase

Flash ignores all other blending modes imported from Fireworks. The blending modes that Flash does not support are Average, Negation, Exclusion, Soft Light, Subtractive, Fuzzy Light, Color Dodge, and Color Burn.

Importing text from Fireworks into Flash

When you import text from Fireworks into Flash 8 or later, the text is imported with the default anti-alias setting of the current document.

If you import the PNG file as a flattened image, you can start Fireworks from Flash and edit the original PNG file (with vector data).

When you import multiple PNG files in a batch, you select import settings one time. Flash uses the same settings for all files in the batch.

***Note:** To edit bitmap images in Flash, convert the bitmap images to vector artwork or break apart the bitmap images.*

- 1 Select File > Import > Import To Stage or Import To Library.
- 2 Select PNG Image from the Files Of Type (Windows) or Show (Macintosh) pop-up menu.
- 3 Navigate to a Fireworks PNG image and select it.
- 4 Click Open.
- 5 Select one of the following for Location:

Import All Page(s) into New Scene(s) Imports all pages in the PNG file as scenes within a movie clip, with all of their frames and layers intact inside the movie clip symbol. A new layer is created that uses the name of the Fireworks PNG file name. The first frame (page) of the PNG document is placed on a keyframe that starts at the last keyframe; all other frames (pages) follow.

Import One Page into Current Layer Imports the selected page (identified in the Page Name pop-up menu) of the PNG file into the current Flash document in a single new layer as a movie clip. The contents of the selected page are imported as a movie clip, with all of their original layer and frame structure intact. If the page movie clip has frames inside it, each frame is a movie clip in itself.

Page Name Specifies the Fireworks page you want to import into the current scene.

6 Select one of the following for File Structure:

Import As Movie Clip And Retain Layers Imports the PNG file as a movie clip, with all of its frames and layers intact inside the movie clip symbol.

Import Page(s) as New Layer(s) Imports the PNG file into the current Flash document in a single new layer at the top of the stacking order. The Fireworks layers are flattened into the single layer. The Fireworks frames are contained in the new layer.

7 For Objects, select one of the following:

Rasterize If Necessary To Maintain Appearance Preserves Fireworks fills, strokes, and effects in Flash.

Keep All Paths Editable Keeps all objects as editable vector paths. Some Fireworks fills, strokes, and effects are lost on import.

8 For Text, select one of the following:

Rasterize If Necessary To Maintain Appearance Preserves Fireworks fills, strokes, and effects in text imported into Flash.

Keep All Paths Editable Keeps all text editable. Some Fireworks fills, strokes, and effects are lost on import.

9 To flatten the PNG file into a single bitmap image, select Import As A Single Flattened Bitmap. When this option is selected, all other options are dimmed.

10 Click OK.

See also

[“Edit a bitmap in an external editor”](#) on page 88

[“Convert a bitmap to a vector graphic”](#) on page 89

[“Break apart a bitmap and create a bitmap fill”](#) on page 89

Working with Illustrator AI files

About Adobe Illustrator AI files

Flash lets you import Adobe® Illustrator® AI files and preserves the majority of the editability and visual fidelity of your artwork. The AI Importer also provides you with a great degree of control in determining how your Illustrator artwork is imported into Flash, letting you specify how to import specific objects into an AI file.

The Flash AI Importer provides the following key features:

- Preserves editability of the most commonly used Illustrator effects as Flash filters.
- Preserves editability of blend modes that Flash and Illustrator have in common.
- Preserves the fidelity and editability of gradient fills.
- Maintains the appearance of RGB (red, green, blue) colors.
- Imports Illustrator Symbols as Flash Symbols.
- Preserves the number and position of Bezier control points.
- Preserves the fidelity of clip masks.
- Preserves the fidelity of pattern strokes and fills.

- Preserves object transparency.
- Converts the AI file layers to individual Flash layers, keyframes, or a single Flash layer. You can also import the AI file as a single bitmap image, in which case Flash flattens (rasterizes) the file.
- Provides an improved copy-and-paste workflow between Illustrator and Flash. A copy-and-paste dialog box provides settings to apply to AI files being pasted onto the Flash stage.

For video tutorials about working with Illustrator and Flash, see:

- Using symbols between Illustrator and Flash: www.adobe.com/go/vid0198
- Using text between Illustrator and Flash: www.adobe.com/go/vid0199

Compatibility between Flash and Illustrator

Certain visual attributes can either not be accurately imported, or, after they are imported, lose their ability to be further edited in the Flash authoring environment. The AI Importer provides you with several options to import and place artwork to best maintain its visual appearance and editability. However, certain visual attributes cannot be preserved. Use the following guidelines to improve the appearance of AI files imported into Flash:

- Flash supports only the RGB color space, and not the CMYK color space, which is common in printing. Flash can convert CMYK images to RGB, however, colors are better preserved if you convert the colors to RGB in Illustrator.
- To preserve the drop shadow, inner glow, outer glow, and Gaussian blur AI effects as editable Flash filters, Flash imports the objects to which they are applied as a Flash movie clip. If you attempt to import an object with these attributes as something other than a movie clip, Flash displays an incompatibility alert, and recommends importing the object as a movie clip.

Working with Illustrator and Flash

You can move Illustrator artwork into the Flash editing environment or directly into Flash Player. You can copy and paste artwork, save files in SWF format, or export artwork directly to Flash. In addition, Illustrator provides support for Flash dynamic text and movie clip symbols. For a video on using Flash and Illustrator together, see www.adobe.com/go/lrvid4099_xp.

You can also use Device Central to see how Illustrator artwork will appear in Flash Player on different handheld devices.

Pasting Illustrator artwork

You can create graphically rich artwork in Illustrator and copy and paste it into Flash simply, quickly, and seamlessly.

When you paste Illustrator artwork into Flash, the following attributes are preserved:

- Paths and shapes
- Scalability
- Stroke weights
- Gradient definitions
- Text (including OpenType fonts)
- Linked images
- Symbols
- Blending modes

In addition, Illustrator and Flash support pasted artwork in the following ways:

- When you select entire top-level layers in Illustrator artwork and paste them into Flash, the layers are preserved along with their properties (visibility and locking).
- Non-RGB Illustrator colors (CMYK, grayscale, and custom) convert to RGB in Flash. RGB colors paste as expected.
- When you import or paste Illustrator artwork, you can use various options to preserve effects (such as drop shadow on text) as Flash filters.
- Flash preserves Illustrator masks.

Exporting SWF files from Illustrator

From Illustrator, you can export SWF files that match the quality and compression of SWF files exported from Flash.

When you export, you can choose from a variety of presets to ensure optimal output, and you can specify how to handle multiple artboards, symbols, layers, text, and masks. For example, you can specify whether Illustrator symbols are exported as movie clips or graphics, or you can choose to create SWF symbols from Illustrator layers.

Importing Illustrator files into Flash

When you want to create complete layouts in Illustrator and then import them into Flash in one step, you can save your artwork in the native Illustrator format (AI) and import it, with high fidelity, into Flash using the File > Import To Stage or File > Import To Library commands in Flash.

If your Illustrator file contains multiple artboards, you select the artboard you want to import from the Import dialog box in Flash, and specify settings for each layer in that artboard. All objects on the selected artboard import as a single layer in Flash. If you import another artboard from that same AI file, the objects from that artboard import as a new layer in Flash.

When you import Illustrator artwork as an AI, EPS, or PDF file, Flash preserves the same attributes as for pasted Illustrator artwork. In addition, when an imported Illustrator file contains layers, you can import them in any of the following ways:

- Convert Illustrator layers to Flash layers
- Convert Illustrator layers to Flash frames
- Convert all Illustrator layers to a single Flash layer

Symbol workflow

Symbol workflow in Illustrator is similar to symbol workflow in Flash.

Symbol creation When you create a symbol in Illustrator, the Symbol Options dialog box lets you name the symbol and set options specific to Flash: movie clip symbol type (which is the default for Flash symbols), Flash registration grid location, and 9-slice scaling guides. In addition, you can use many of the same symbol keyboard shortcuts in Illustrator and Flash (such as F8 to create a symbol).

Isolation mode for symbol editing In Illustrator, double-click a symbol to open it in isolation mode for easier editing. In isolation mode, only the symbol instance is editable—all other objects on the artboard are dimmed and unavailable. After you exit isolation mode, the symbol in the Symbols panel, and all instances of that symbol are updated accordingly. In Flash, symbol-editing mode and the Library panel work in a similar fashion.

Symbol properties and links Using the Symbols panel or the Control panel, you can easily assign names to symbol instances, break links between instances and symbols, swap a symbol instance with another symbol, or create a copy of the symbol. In Flash, the editing features in the Library panel work in a similar way.

Static, dynamic, and input text objects

When you bring static text from Illustrator into Flash, Flash converts the text to outlines. In addition, you can set up your text in Illustrator as dynamic text. Dynamic text enables you to edit text content programmatically in Flash, and easily manage projects that require localization in multiple languages.

In Illustrator, you can specify individual text objects as static, dynamic, or input text. Dynamic text objects in Illustrator and Flash have similar properties. For example, both use kerning that affects all characters in a text block rather than individual characters, both anti-alias text the same way, and both can be linked to an external XML file containing text.

About importing AI files to the Flash library

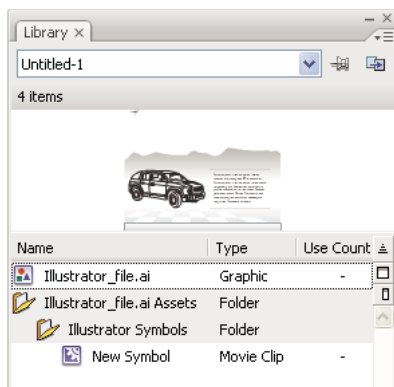
Importing an AI file into the library is similar to importing to the Stage, the difference being that the entire AI file is encapsulated as a Flash symbol. The content is imported into the library and organized by the layering and grouping structure of the AI file.

When you import an AI file into the library, the root folder uses the name of the AI file. After the AI file is imported into the library, you can change the name of the root folder, or move the layers out of the folder.

***Note:** The Library panel orders the contents of the imported AI file alphabetically. The hierarchical grouping and folder structure remains the same, but the library reorders them alphabetically.*

When converting AI layers to keyframes, the AI file is imported as a movie clip; converting the AI layers to Flash layers, or as a single Flash layer, imports the AI file as a graphic symbol. The resulting movie clip or graphic symbol contains all of the content of the AI file imported to its timeline, as if the content were imported to the Stage. Almost all movie clips have a bitmap or other asset associated with them. To minimize confusion and naming conflicts, these assets are stored in an Assets folder in the same folder as the movie clip.

***Note:** When you import to the library, the AI file's contents are imported to the movie clip's timeline, not the main Flash timeline.*



The Library panel after importing an AI file

AI File Importer preferences

The Flash Preferences dialog box lets you set import preferences for AI files and the AI File Import dialog box. The preferences you specify for importing AI files affect the options the AI Import dialog box is initially populated with for the Illustrator object types.

Note: To override the preferences specified for different layer types on an object by object basis, use the AI Import dialog box. Select the layer, object, or group to change its import options, and specify the necessary options.

General Preferences that affect how the AI Importer will respond when importing AI files:

- **Show Import Dialog Box** Specifies that the AI File Import Dialog dialog box appears.
- **Exclude Objects Outside Crop Area** Excludes objects on the Illustrator canvas which fall outside the artboard or crop area.
- **Import Hidden Layers** Specifies that hidden layers be imported by default.

Import Text As Lets you specify the following import preferences for text objects:

- **Editable Text** Specifies that imported Illustrator text is editable Flash text. The appearance of the text may be compromised to maintain the editability of the text.
- **Vector Outlines** Converts text to vector paths. Use this option to preserve the visual appearance of the text. Some visual effects might be sacrificed—such as unsupported blend modes and filters—but visual attributes such as text on a path are maintained if the text is imported as a movie clip. The text itself is no longer editable, but opacity and compatible blend modes maintain their editability.

Note: To preserve the drop shadow, inner glow, outer glow, and Gaussian blur that AI effects applied to text as editable Flash filters, select *Create Movie Clip Import* to import the text as a movie clip.

- **Bitmaps** Rasterizes the text into a bitmap to preserve the exact appearance of the text as it was in Illustrator. If filters or other effects are applied that are not compatible with Flash, importing the text as a bitmap preserves the visual appearance. Rasterized text is no longer editable.
- **Create Movie Clips** Specifies that text objects be imported inside a movie clip. To maintain supported blend modes, AI effects, and transparency of less than 100% between Illustrator and Flash, specify that the text object be imported as a movie clip.

Import Paths As Lets you specify the following path import preferences:

- **Editable Paths** Creates an editable vector path. Supported blend modes, effects, and object transparency are preserved, but attributes not supported in Flash are discarded.
- **Bitmaps** Rasterizes the path into a bitmap to preserve the exact appearance of the path in Illustrator. A rasterized image is no longer editable.
- **Create Movie Clips** Specifies that path objects be imported inside a movie clip.

Images Lets you specify the import preference for images:

- **Flatten Bitmaps To Maintain Appearance** Rasterizes the image into a bitmap to preserve the appearance of blending modes and effects that are not supported in Flash. A rasterized image is no longer editable.
- **Create Movie Clips** Specifies that images be imported inside a movie clip.

Groups Lets you specify the import preferences for groups:

- **Import As Bitmaps** Rasterizes the group into a bitmap to preserve the appearance of the objects as they appeared in Illustrator. After a group is converted to a bitmap, the objects in it cannot be selected or renamed.
- **Create Movie Clips** Specifies that all objects in the group be encapsulated into a single movie clip.

Layers Lets you specify the import preferences for layers:

- **Import As Bitmaps** Rasterizes the layer into a bitmap to preserve the appearance of the objects as they appeared in Illustrator.
- **Create Movie Clips** Specifies that the layer is encapsulated into a movie clip.

Movie Clip Registration Specifies a global registration point for movies that are created. This setting applies to the registration point for all object types. This option can be changed on an object by object basis in the AI File Import dialog box; this is the initial setting for all object types. For more information on movie clip registration, see “[Edit symbols](#)” on page 150.

See also

“[Set preferences in Flash](#)” on page 37

Import Adobe Illustrator files

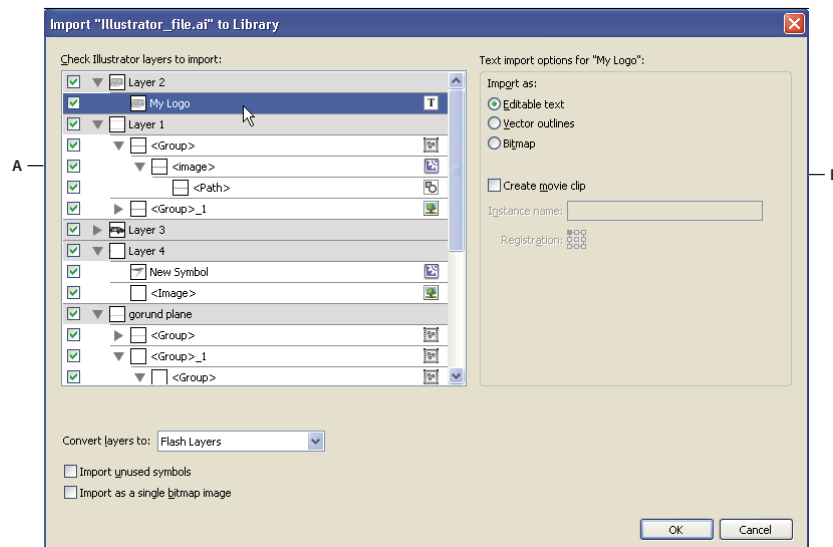
Flash can import Illustrator AI files in version 10 or earlier. If the raster file in Illustrator is linked, only JPEG, GIF, and PNG are imported with native format preserved. All other files are converted to PNG format in Flash. Additionally, conversion to PNG depends on the version of QuickTime installed.

***Note:** The AI Importer was developed to import AI files created with Illustrator CS3 and later. While there are no known issues importing AI files created in earlier versions of Illustrator, we recommend importing AI files created using Illustrator CS3 and later. If you encounter issues importing AI files created in an earlier version of Illustrator, open the file in Illustrator CS3 or later, save the AI file as a CS3 or later compatible file, and re-import it into Flash. To import an Illustrator EPS or Adobe® Acrobat® PDF file, open the file in Illustrator CS3 or later, save it as a CS3 or later compatible AI file, and import the resulting AI file into Flash.*

For video tutorials about working with Illustrator and Flash, see:

- Importing Illustrator files: www.adobe.com/go/vid0197
- Using symbols between Illustrator and Flash: www.adobe.com/go/vid0198
- Using text between Illustrator and Flash: www.adobe.com/go/vid0199

In certain instances, you may need to ungroup all the Illustrator objects on all layers when Flash cannot group items on import. After all the objects are ungrouped, they can be manipulated like any other Flash object.



*Illustrator AI Import dialog box
 A. Layers panel B. Import options available for the selected object*

Choose from the following options when importing Adobe Illustrator files:

- Convert layers to Flash layers or keyframes, or single Flash layers.
- Import text as editable text, vector outlines, or bitmaps.
- Import as a single bitmap image. Choosing this option imports the AI file as a single bitmap image and disables the layers list and import options within the AI Import dialog box.

1 Select File > Import To Stage or Import To Library.

2 Navigate to the AI file to import, select it, and click OK. The Import Illustrator Document To Stage or The Import Illustrator Document To Library dialog box appears.

This dialog box provides options to import the Illustrator file. Depending on the types of objects in the Illustrator file you're importing, the options available to you vary.

3 (Optional) To generate a list of items in the AI file that are incompatible with Flash, click Incompatibility Report. The Incompatibility Report button appears only if incompatibilities exist with Flash in the AI file.

The incompatibility report analyzes possible incompatibilities between Illustrator and Flash. The Import Options area (next to the Alert button) of the AI Import dialog box displays recommendations to obtain the maximum compatibility for any incompatible item.

The incompatibility report includes an Apply Recommended Import Settings checkbox. If checked, Flash automatically applies the recommended import options to any incompatible objects within the AI file. The exceptions to this are when the AI document is larger in size than that supported by Flash, and when the AI document uses the CMYK color mode. To correct either of these incompatibilities, reopen the document in Illustrator CS3, and adjust the size of the document, or change the color mode to RGB.

4 (Optional) In the AI Import dialog box, select layers, groups, and individual objects, and choose how to import each item. For information on the import options available for the different types of Illustrator layers and objects, see [“Illustrator object import options”](#) on page 76.

5 For Convert Layers To, select one of the following:

Flash Layers Converts each layer in the imported document to a layer in the Flash document.

Keyframes Converts each layer in the imported document to a keyframe in the Flash document.

Single Flash Layer Converts all layers in the imported document to a single flattened layer in the Flash document.

6 For the remaining options, select from the following:

Place Objects At Original Position The contents of the AI file retain the exact position that they had in Illustrator. For example, if an object was positioned at X = 100 Y = 50 in Illustrator, it assumes the same coordinates on the Flash Stage.

If this option is not selected, the imported Illustrator layers are centered within the current view. The items in the AI file remain relative to each other when imported; however, all objects are centered as one block in the current view. This feature can be useful if you are zoomed in to an area of the Stage and are importing a specific object for that area of the Stage. If you imported the object using the original coordinates, you might not see it imported, because it might be placed outside of the current stage view.

Note: When importing the AI file into the Flash library, this option is not available.

Set Stage Size To Same Size As Illustrator Artboard The Flash stage size resizes to the same size as the Illustrator artboard (or active crop area) used to create the AI file. This option is not selected by default.

Note: When importing the AI file into the Flash library, this option is not available.

Import Unused Symbols Any symbols in the AI file's library that have no instances on the artboard are imported into the Flash library. If this option is not selected, the unused symbols are not imported into Flash.

Import As A Single Bitmap Image Imports the AI file as a single bitmap image and disables the layers list and import options within the AI Import dialog box.

7 Click OK.

See also

[“Import Photoshop PSD files”](#) on page 80

[“Exporting Images and Graphics”](#) on page 412

Illustrator object import options

Layers manage all the items that make up Illustrator artwork. By default, all items are organized in a single, parent layer. You can import all items in a given parent layer as a single, flattened bitmap, or, individually select each object and specify import options specific to the type of object it is (text, path, group, and so on). The AI Importer provides options to select layers in the artwork you are importing and specify individual import options based on preserving the visual appearance of an object, or its editability in Flash.

Correcting incompatible graphic effects

- 1 To generate an incompatibility report, click Incompatibility Report. The incompatibility report lists items in the AI file that are incompatible with Flash.
- 2 Select Change Import Settings To Resolve Object Incompatibilities. Many incompatibilities between Illustrator and Flash can be automatically corrected using the incompatibility report and the import recommendations suggested in the Import options area of the AI Import dialog box.

Select individual objects

- 1 Select the object to specify import options for. Illustrator objects that you can select include layers, groups, individual paths, text, and images.
- 2 Review the import options available for the type of object you've selected in the object options section of the dialog box. Note whether any incompatibilities are listed and what the recommended correction is to import the object.
- 3 Select the desired import options, and select another object to specify import options for, or click OK.

Selecting layers







The Layers panel lists the objects in a document. By default, every Illustrator document contains at least one layer, and each object in a given file is listed under that layer.

When an item in the Layers panel contains other items, a triangle appears to the left of the item's name. To show or hide the content, click the triangle. If no triangle appears, the layer contains no additional items.

Note: You can expand or collapse all groups and layers using the AI Importer context menu. Right-click to display the context menu and select Expand All or Collapse All.

Selection column Controls whether items are selected for import or not. If an item is checked, you can select that layer and specify import options; if Edit is not checked, the layer is dimmed and you cannot specify any import setting for the item in that layer.

Object Type column An icon indicates what the Flash object type located on that layer will be when imported and, if visible, that the item is selected. The object types are:

- Text 
- Path 
- Group 
- Movie clip 
- Graphic symbol 
- Image 

Text import options

Flash lets you import text as editable text, vector outlines, or a flattened bitmap. To maintain supported blend modes, AI effects, and transparency of less than 100% between Illustrator and Flash, import text as a movie clip. Importing text as a movie clip preserves the editability of compatible visual effects.

Editable Text By default, Illustrator text imports as editable Flash text. The appearance of the text can be compromised to maintain the editability of the text.

Vector Outlines Converts text to vector paths. Use this option to preserve the visual appearance of the text. Some visual effects might be sacrificed—such as unsupported blend modes and filters—but visual attributes such as text on a path are maintained if the text is imported as a movie clip. The text itself is no longer editable, but opacity and compatible blend modes maintain their editability.

Note: To preserve the drop shadow, inner glow, outer glow, and Gaussian blur that AI effects applied to text as editable Flash filters, select Create Movie Clip Import to import the text as a movie clip.

Bitmap Rasterizes the text into a bitmap to preserve the exact appearance of the text as it was in Illustrator. If filters or other effects are applied that are not compatible with Flash, importing the text as a bitmap preserves the visual appearance. Rasterized text is no longer editable.

Path import options

A path is the line that results from drawing in Illustrator. A path is either open, like an arc, or closed, like a circle. For an open path, the starting and ending anchor points for the path are called endpoints. Editable paths can be imported into Flash, however, if certain blend modes, filters, or other effects are applied to the path, those effects might not be compatible with Flash.

Bitmap Rasterizes the path into a bitmap to preserve the exact appearance of the path in Illustrator. A rasterized image is no longer editable.

Editable Path Creates an editable vector path. Supported blend modes, effects, and object transparency are preserved, but attributes not supported in Flash are discarded.

Image import options

Bitmap images are the most common electronic medium for continuous-tone images, such as photographs or digital paintings. Illustrator creates bitmap effects by using filters, effects, and graphic styles. While many of these effects are compatible with Flash, some might need to be flattened—or rasterized—to preserve their visual appearance.

Note: If a raster file in Illustrator is linked, only JPEG, GIF, and PNG are imported with their native format preserved. All other file types are converted to the PNG format in Flash. Additionally, the conversion (to PNG) depends on the version of QuickTime® installed on your computer.

Flatten Bitmap To Maintain Appearance Rasterizes the image into a bitmap to preserve the appearance of blending modes and effects that are not supported in Flash. A rasterized image is no longer editable.

Create Movie Clip Imports Illustrator images as movie clips.

Group import options

Groups are collections of graphic objects that are treated as a single unit. Grouping lets you move or transform a number of objects without affecting their attributes or relative positions. For example, you might group the objects in a logo design so that you can move and scale the logo as one unit. Groups can also be nested. That is, they can be grouped in other objects or groups to form larger groups.

Groups appear as <Group> items in the Import panel. When an item such as a group contains other items, a triangle appears to the left of the item's name. Click the triangle to show or hide the contents of the group. If no triangle appears, the item contains no additional items.

Import As Bitmap Rasterizes the group into a bitmap to preserve the appearance of the objects as they appeared in Illustrator. After a group is converted to a bitmap, the objects in it cannot be selected or renamed.

Create Movie Clip Encapsulates all objects in the group into a single movie clip.

Copy and pasting between Illustrator and Flash

If you copy and paste (or drag-and-drop) artwork between Illustrator and Flash, the Paste dialog box appears, which provides import settings for the AI file being copied (or pasted).

Paste As Bitmap Flattens the file being copied into a single bitmap object.

Paste Using AI File Importer Preferences Imports the file using the AI file import setting specified in Flash Preferences (Edit > Preferences).

Apply Recommended Import Settings To Resolve Incompatibilities Enabled by default when Paste Using AI File Importer Preferences is selected. Automatically fixes any incompatibilities detected in the AI file.

Maintain Layers Enabled by default when Paste Using AI File Importer Preferences is selected. Specifies that layers in the AI file be converted to Flash layers (the same as if you selected Convert To Flash Layers from the AI Import dialog box). If deselected, all layers are flattened into a single layer.

Working with Photoshop PSD files

Working with Photoshop and Flash

You can use Adobe® Photoshop® and Adobe® Flash® together to create visually compelling web-based applications, animations, or interactive messaging elements. Photoshop lets you create still images and artwork, providing a high degree of creative control. Flash lets you bring those still images together and incorporate them into interactive Internet content.

The Photoshop drawing and selection tools provide you with a greater degree of creative control than the tools found in Flash. If you have to create complex visual images, or retouch photographs for use in interactive presentations, use Photoshop to create your artwork, and then import the finished images into Flash.

Importing still images into Flash

Flash can import still images in many formats, but you usually use the native Photoshop PSD format when importing still images from Photoshop into Flash.

When importing a PSD file, Flash can preserve many of the attributes that were applied in Photoshop, and provides options for maintaining the visual fidelity of the image and further modifying the image. When you import a PSD file into Flash, you can choose whether to represent each Photoshop layer as a Flash layer, individual keyframes, or a single, flattened image. You can also encapsulate the PSD file as a movie clip.

Exchanging movies

You can exchange QuickTime video files between Photoshop and Flash. For example, you can render a QuickTime movie directly from Photoshop and then import it into Flash, converting it into an FLV (video for Adobe Flash Player) file, which can be played in Flash® Player.

When you use Photoshop to modify video footage, you can paint nondestructively on its frames. When you save a Photoshop file with a video layer, you are saving the edits that you made to the video layer, not edits to the footage itself.

Note: When you import a QuickTime video file from Photoshop to Flash, use the Import Video dialog box (File > Import Video). Using the Photoshop PSD import feature to import video only imports the first frame of a video file.

You can also export Flash documents as QuickTime video, and import them into Photoshop where you can paint nondestructively on the video frames. For example, you can create an animated sequence in Flash, export the Flash document as a QuickTime video, and then import the video into Photoshop.

Color

Flash works internally with colors in the RGB or HSB (hue, saturation, brightness) color space. Although Flash can convert CMYK images to RGB, you should create Photoshop artwork in RGB. Before you import CMYK artwork from Photoshop into Flash, convert the image to RGB in Photoshop.

See also

“[Import Photoshop PSD files](#)” on page 80

“[Export QuickTime](#)” on page 416

About importing Photoshop PSD files

Flash lets you import Photoshop PSD files and preserves most of your artwork's data. The PSD Importer also lets you control how your Photoshop artwork is imported into Flash, letting you specify how to import specific objects in a PSD file, as well as specifying that the PSD file be converted to a Flash movie clip.

The Flash PSD Importer provides the following key features:

- PSD files imported into Flash maintain their color fidelity from Photoshop.
- Preserves editability of blend modes that both Flash and Photoshop have in common.
- Smart Objects in the PSD file are rasterized, and imported into Flash as bitmaps which preserves object transparency.
- Converts the PSD file layers to individual Flash layers or keyframes or imports the PSD file as a single bitmap image, in which case Flash flattens (rasterizes) the file.
- Drag-and-drop from Photoshop to Flash invokes the PSD file importer, letting you choose how to import your Photoshop artwork.

For a video tutorial about designing websites with Photoshop and Flash, see www.adobe.com/go/vid0201.

Compatibility between Flash and Photoshop

Certain visual attributes can either not be accurately imported, or, after they are imported, lose their ability to be further edited in the Flash authoring environment. The PSD Importer provides you with several options to import and place artwork to best maintain its visual appearance and editability. However, certain visual attributes cannot be preserved. Use the following guidelines to improve the appearance of PSD files imported into Flash:

- Flash supports only the RGB color space, and not the CMYK color space, which is common in printing. Flash can convert CMYK images to RGB; however, colors are better preserved if you convert CMYK colors to RGB in Photoshop.
- Flash can import the following Photoshop blend modes, and maintain their editability: Normal, Darken, Multiply, Lighten, Screen, Hard Light, Difference, and Overlay.

If you use a blend mode that Flash does not support, you can rasterize the layer to maintain its visual appearance, or remove the blend mode from the layer.

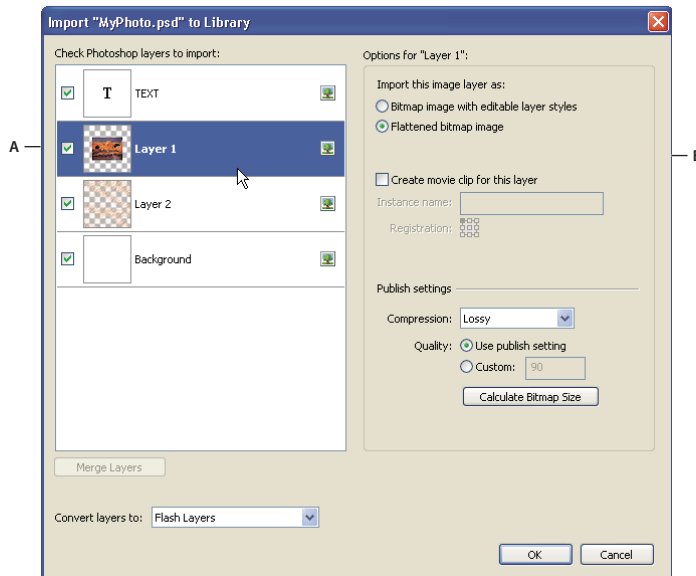
- Flash cannot import Photoshop Smart Objects as editable objects. To preserve the visual attributes of Smart Objects, they are rasterized and imported into Flash as bitmaps.
- Flash can only import the first frame of Photoshop Video layers.
- Image and Fill layers are always rasterized when imported into Flash.
- PNG objects in Photoshop are converted to JPG files when imported into Flash. Transparency in the original PNG is preserved in the resulting JPG file.
- Photoshop does a better job of scaling bitmap images than Flash. If you know that you plan to scale a bitmap that is coming from Photoshop into Flash, scale the bitmap in Photoshop before importing it into Flash.
- When importing objects containing transparent areas as flattened bitmaps, any objects on the layers behind the transparent portion of the object will be visible through the transparent area—assuming the objects behind the object containing transparency are also being imported. To prevent this, import only the transparent object as a flattened bitmap.

To import multiple layers and maintain transparency without having any visible remnants of the layers behind the transparency, import the PSD file using the Bitmap Image With Editable Layer Styles option. This will encapsulate the imported objects as a movie clip, and use the movie clip's transparency. This is particularly useful if you need to animate the different layers in Flash.

Import Photoshop PSD files

Photoshop format (PSD) is the default Photoshop file format. Flash can directly import PSD files and preserve many Photoshop features, retaining the image quality and editability of the PSD file in Flash. You can also flatten PSD files when importing them, creating a single bitmap image file that retains the image's visual effects but removes the hierarchical layer information native to the PSD file format.

For a video tutorial about designing websites with Photoshop and Flash, see www.adobe.com/go/vid0201.



The PSD Import dialog box
A. Layers in the PSD file being imported B. Import options available for the selected layer or object

- 1 Select File > Import To Stage or Import To Library.
- 2 Navigate to the Adobe Photoshop PSD file to import, select it, and click OK.
- 3 (Optional) In the PSD Import dialog box, select layers, groups, and individual objects and choose how to import each item.
- 4 For Convert Layers To, select one of the following:

Flash Layers All selected layers in the Select Photoshop layers list are placed on their own layer. Each layer is labeled with the name of the layer in the Photoshop file. The layers in Photoshop are objects on the individual layers. The objects also have the name of the layer in Photoshop when put into the Library panel.

Keyframes All selected layers in the Select Photoshop layers list are placed in individual keyframes on a new layer. The new layer is named for the Photoshop file (for example, myfile.psd). The layers in Photoshop become objects on the individual keyframes. The objects also have the name of the layer in Photoshop when put into the Library panel.

- 5 For the remaining options, select from the following:

Place Layers At Original Position The contents of the PSD file retain the exact position that they had in Photoshop. For example, if an object was positioned at X = 100 Y = 50 in Photoshop, it assumes the same coordinates on the Flash Stage.

If this option is not selected, the imported Photoshop layers are centered on the Stage. The items in the PSD file remain relative to each other when imported; however, all objects are centered as one block in the current view. This feature can be useful if you are zoomed in to an area of the Stage and are importing a specific object for that area of the Stage. If you imported the object using the original coordinates, you might not see it imported, because it might be placed outside of the current stage view.

Note: When you import the PSD file to the Flash library, this option is not available.

Set Stage Size To Same Size As Photoshop Canvas The Flash Stage size resizes to the same size as the Photoshop document size (or active crop area) used to create the PSD file. This option is not selected by default.

Note: When you import the PSD file to the Flash library, this option is not available.

6 Click OK.

Importing PSD files to the Flash library

Importing a PSD file into the library is similar to importing to the Stage. When you import a PSD file into the library, the root folder uses the name of the PSD file. After the PSD file is imported into the library, you can change the name of the root folder, or move the layers out of the folder.

***Note:** The library orders the contents of the imported PSD file alphabetically. The hierarchical grouping and folder structure remains the same, but the library reorders them alphabetically.*

A movie clip is created that contains all of the content of the PSD file imported to its timeline, as if the content were imported to the Stage. Almost all movie clips have a bitmap or other asset associated with them. To minimize confusion and naming conflicts, these assets are stored in an Assets folder in the same folder as the movie clip.

***Note:** When you import to the library, the PSD file's contents are imported to the movie clip's timeline, not the main Flash timeline.*

Photoshop Import options

When you import a Photoshop PSD file that contains multiple layers, you can set the following options:

Layer Comp If the Photoshop file contains layer comps, you can specify which version of the image to import. A *layer comp* is a snapshot of a state of the Photoshop Layers palette. Layer comps record three types of layer options, all of which are imported into Flash:

- **Layer visibility:** whether a layer is displayed or hidden.
- **Layer position** in the document.
- **Layer appearance:** whether a layer style is applied to the layer and the layer's blending mode.

If no layer comps are present, this pop-up menu is hidden. Flash supports all aspects of the layer comp's fidelity, including visibility, position, and layer style.

Select Photoshop Layers Lists all layers, groups, and layer effects in an image. To select which layers to import, use the options to the left of the layer's thumbnail. By default, all layers visible in Photoshop are checked and invisible layers are not checked.

***Note:** Adjustment layers do not have a compatible layer type in Flash. For this reason, when importing as a flattened bitmap, the visual effect of the adjustment layer is applied to maintain the appearance of the Photoshop layer being imported. If you've selected other import options, the adjustment layers will not be applied.*

Merge Layers Merge (or collapse) two or more layers into a single bitmap, and then import the resulting single bitmap object rather than individual objects. You can only merge layers that are on the same level and the selection must be contiguous. For example, you cannot select an item inside a folder and an item outside a folder and merge them. Instead, select the folder and the item outside the folder to merge them. You cannot Control-click individual items that are out of sequence and merge them.

***Note:** If you select a merged bitmap object, the Merge Layers button changes to the Separate button. To separate any merged bitmap objects you create, select the resulting single bitmap, and click the Separate button.*

Importing text objects

Text objects are text layers in Photoshop. Choose how to import text into Flash.

Editable Text Creates an editable text object from the text on the selected Photoshop layer. The appearance of the text might be compromised to maintain the editability of the text. If you import the text as a movie clip, the movie clip contains an editable text object.

Note: When importing editable text into the library, it must be inside a movie clip. Only movie clips, bitmaps, and graphic symbols can be stored in the library. When Editable Text is selected as an option for a text layer being imported into the library, it is automatically imported as a graphic symbol.

Vector Outlines Converts text to vector paths to preserve the visual appearance of the text. The text itself is no longer editable, but opacity and compatible blend mode maintain their editability. If this option is selected, the object must be converted to a movie clip.

Flattened Bitmap Image Rasterizes the text into a bitmap to preserve the exact appearance of the text layer in Photoshop. Rasterized text is no longer editable.

Note: When importing text on a path you must import it as a flattened bitmap image to preserve the visual fidelity of the object.

Importing shape objects

A Shape Layer object is an object that was originally a shape layer in Photoshop or an image layer with a vector clipping mask on it.

Editable Paths And Layer Styles Creates an editable vector shape with a bitmap clipped inside the vector. Supported blend modes, filters, and opacity are maintained. Unsupported blend modes that cannot be reproduced in Flash are removed. The object must be converted to a movie clip.

Flattened Bitmap Image Rasterizes the shape into a bitmap to preserve the exact appearance of the shape layer in Photoshop. A rasterized image is no longer editable.

Importing image or fill layers

If the image or fill layer is associated with a vector mask, it is treated as a shape layer object.

Bitmap Image With Editable Layer Styles Creates a movie clip with a bitmap inside. Supported blend modes, filters, and opacity are maintained. Unsupported blend modes that cannot be reproduced in Flash are removed. The object must be converted to a movie clip.

Flattened Bitmap Image Rasterizes the image into a bitmap to preserve the exact appearance of the image or fill layer in Photoshop.

Importing merged bitmap objects and objects in a merged bitmap

A *merged bitmap* is an object that contains more than one Photoshop layer that is flattened (or merged) into a single bitmap when imported into Flash. Objects in a merged bitmap represent layers in Photoshop. To create a merged bitmap, select two or more layers, and select the Merge Layers button.

Importing multiple objects of different types

If you import multiple objects of different types, Flash only lets you import the selected objects with the import options they share in common, such as Create Movie Clip and Registration.

Importing multiple objects of the same type

If you import multiple objects of the same type, the import options displayed are the same as if a single object of that type was selected. If the objects do not share the same attributes, the import options displayed are in an indeterminate state, and your results may not be as expected.

Importing a Group folder

When you import a group folder, you can import it as a movie clip, or place each layer in the group on its own layer or keyframe on the timeline.

If you select Import As Movie Clip, each layer in the group folder is placed on a layer in a movie clip, which is then placed on its own layer or keyframe on the timeline. The movie clip uses the same name as that group folder had in Photoshop, and if you import the movie clip to a Flash layer, the layer uses the same name.

If you do not place the group in a movie clip, each layer is converted to the type that is currently set for it, and each layer in the group is imported to its own Flash layer. The Flash layers are named for the individual layers in the PSD file.

Importing and merging layers

The PSD Importer lets you merge two or more layers into a merged bitmap that is imported as a single bitmap file rather than individual objects.

The layers you select to create a merged bitmap must be a continuous range of two or more layers at the same level. For example, you cannot select one layer inside a group and a layer outside that group, and merge them. Instead, you must select the entire group and the separate layer.

Setting publish options

The publish settings in the PSD Importer let you specify the degree of compression and document quality to apply to the image when publishing the Flash document as a SWF file. These settings take effect only when you publish the document as a SWF file and have no effect on the image when you import it to the Flash Stage or library.

Compression Lets you choose either lossy or lossless compression formats:

- **Lossy** Lossy (JPEG) compresses the image in JPEG format. To use the default compression quality specified for the imported image, select Use Publish Setting. To specify a new quality compression setting, select the Custom option and enter a value between 1 and 100 in the Quality text field. (A higher setting preserves greater image integrity but yields a larger file size.)
- **Lossless** Lossless (PNG/GIF) compresses the image with lossless compression, in which no data is discarded from the image.

Note: Use lossy compression for images with complex color or tonal variations, such as photographs or images with gradient fills. Use lossless compression for images with simple shapes and relatively few colors.

Calculate Bitmap Size Determines the number of bitmaps that are created for a given layer based on your import selections and the compressed size in kilobytes of the resulting bitmaps on the layer. For example, if you select a layer with a drop shadow and a blur and maintain layer styles, the Calculate Bitmap Size information states that three bitmaps will result from the import—one for each filter effect and another for the image itself. To calculate the size of all the bitmaps to be imported, select all of the layers and click Calculate Bitmap Size.

Photoshop file importer preferences

The Flash Preferences dialog box lets you set import preferences for Photoshop PSD files. The preferences you specify for importing PSD files affect the options the PSD Import dialog box is initially populated with for the Photoshop layer types.

Note: To override the preferences specified for different layer types on an object by object basis, use the PSD Import dialog box. Select the layer to change import options for, and specify the necessary options.

See also

[“Set preferences in Flash”](#) on page 37

Import Image Layers Preferences

These options specify how the import options for image layers will initially be set. You can import image layers as any of the following:

Bitmap Images With Editable Layer Styles Creates a movie clip with a bitmap nested inside. Specifying this option maintains supported blend modes and opacity, but other visual attributes that cannot be reproduced in Flash will be removed. If this option is selected, the object must be converted into a movie clip.

Flattened Bitmap Images Rasterizes any text into a flattened bitmap image to maintain the exact appearance the text layer had in Photoshop.

Create Movie Clips Specifies that the image layers be converted to movie clips when imported into Flash. This option can be changed in the PSD Import dialog box on a layer by layer basis if you do not want all of the image layers to be converted to movie clips.

Text layer import preferences

These options specify how the import options for text layers will initially be set.

Editable Text Creates an editable text object from the text on the Photoshop text layer. The appearance of the text will be sacrificed to maintain the editability of the text. If this option is selected, the object must be converted into a movie clip.

Vector Outlines Vectorizes the text into paths. The appearance of the text may be altered, but visual attributes will be maintained. If this option is selected, the object must be converted into a movie clip.

Flattened Bitmap Images Rasterize the text to maintain the exact appearance the text layer had in Photoshop.

Create Movie Clips Automatically converts the text layer to a movie clip when imported into Flash. This option can be changed in the PSD Import dialog box on an object by object basis if you do not want all of the text layers to be converted to movie clips. This option is required when either Editable Text or Vector Outlines is selected.

Shape layer import preferences

These options specify how the import options for shape layers will initially be set.

Editable Paths And Layer Styles This option will create an editable vector shape with a bitmap clipped inside the vector shape. Supported blend modes and opacity will also be maintained with this option, but other visual attributes that cannot be reproduced in Flash will be sacrificed. If this option is selected, then the object must be converted into a movie clip.

Flattened Bitmap Images This option will rasterize the shape and maintain the exact appearance the shape layer had in Photoshop.

Create Movie Clips This option will set the shape layers to be converted to a movie clip when imported into Flash. This option can be changed on an object by object basis if you do not want some shape layers to be converted to movie clips. This option is disabled if the Maintain Editable Paths and Layers Styles checkbox is checked.

Layer group import preferences

This option specifies how the options for layer groups will initially be set.

Create Movie Clips Specifies that all groups be converted to a movie clip when imported into Flash. This can be changed on an object by object basis if you do not want some layer groups to be movie clips.

Merged bitmap import preferences

This option specifies how the import options for merged bitmaps will initially be set.

Create Movie Clips This option will set the merged bitmaps to be converted to a movie clip when imported into Flash. This option can be changed on an object by object basis if you do not want some merged bitmaps to be converted to movie clips. This option is disabled if the Maintain Editable Paths And Layers Styles checkbox is checked.

Movie clip registration import preferences

Specifies a global registration point for movies that are created. This setting applies to the registration point for all object types. This option can be changed on an object by object basis in the PSD Import dialog box; this is the initial setting for all object types. For more information on movie clip registration, see “[Edit symbols](#)” on page 150.

Publish settings for imported images

The publish settings preferences for a FLA file let you specify the degree of compression and document quality to apply to images when publishing the Flash document as a SWF file. These settings take effect only when you publish the document as a SWF file, and have no effect on the image when you import it to the Flash stage or library.

Compression Lets you choose either lossy or lossless compression formats:

- **Lossy** Lossy (JPEG) compresses the image in JPEG format. To use the default compression quality specified for the imported image, select Use Publish Setting. To specify a new quality compression setting, select the Custom option and enter a value between 1 and 100 in the Quality text field. (A higher setting preserves greater image integrity but yields a larger file size.)
- **Lossless** Lossless (PNG/GIF) compresses the image with lossless compression, in which no data is discarded from the image.

Note: Use lossy compression for images with complex color or tonal variations, such as photographs or images with gradient fills. Use lossless compression for images with simple shapes and relatively few colors.

Quality Lets you set a quality level for the compression.

- **Use Publish Setting** Applies the current JPEG Quality setting from the Publish Settings.
- **Custom** Lets you specify a separate specific quality setting.

Imported bitmaps

Work with imported bitmaps

When you import a bitmap into Flash, you can modify that bitmap and use it in your Flash document in a variety of ways.

If a Flash document displays an imported bitmap at a size larger than the original, the image may be distorted. To be sure that images are displayed properly, preview imported bitmaps.

When you select a bitmap on the Stage, the Property inspector displays the bitmap's symbol name and its pixel dimensions and position on the Stage. Using the Property inspector, you can *swap* an instance of a bitmap—that is, replace the instance with an instance of another bitmap in the current document.

Display the Property inspector with bitmap properties

- 1 Select an instance of a bitmap on the Stage.
- 2 Select Window > Properties.

Replace an instance of a bitmap with an instance of another bitmap

- 1 Select a bitmap instance on the Stage.
- 2 Select Window > Properties, and click Swap.
- 3 Select a bitmap to replace the one currently assigned to the instance.

Set bitmap properties

You can apply anti-aliasing to an imported bitmap to smooth the edges in the image. You can also select a compression option to reduce the bitmap file size and format the file for display on the web.

- 1 Select a bitmap in the Library panel and click the Properties button at the bottom of the Library panel.
- 2 Select Allow Smoothing. Smoothing improves the quality of bitmap images when they are scaled.
- 3 For Compression, select one of the following options:

Photo (JPEG) Compresses the image in JPEG format. To use the default compression quality specified for the imported image, select Use Document Default Quality. To specify a new quality compression setting, deselect Use Document Default Quality and enter a value between 1 and 100 in the Quality text field. (A higher setting preserves greater image integrity but yields a larger file size.)

Lossless (PNG/GIF) Compresses the image with lossless compression, in which no data is discarded from the image.

Note: Use Photo compression for images with complex color or tonal variations, such as photographs or images with gradient fills. Use lossless compression for images with simple shapes and relatively few colors.

- 4 To determine the results of the file compression, click Test. To determine if the selected compression setting is acceptable, compare the original file size to the compressed file size.
- 5 Click OK.

Note: JPEG Quality settings that you select in the Publish Settings dialog box do not specify a quality setting for imported JPEG files. Specify a quality setting for each imported JPEG file in the Bitmap Properties dialog box.

Import a bitmap at runtime

To add bitmaps to a document at runtime, use the `ActionScript® 2.0` or the `ActionScript 3.0 BitmapData` command. To do so, specify a linkage identifier for the bitmap. For more information, see [Assigning linkage to assets in the library in *Learning ActionScript 2.0 in Adobe Flash*](#) or [Exporting library symbols for ActionScript in *Programming ActionScript 3.0*](#).

- 1 Select the bitmap in the Library panel.
- 2 Do one of the following:
 - Select Linkage from the Panel menu in the upper-right corner of the panel.

- Right-click (Windows) or Control-click (Macintosh) the bitmap name in the Library panel, and select Properties from the context menu.

If the Linkage properties aren't visible in the Properties dialog box, click Advanced.

- 3 For Linkage, select Export For ActionScript.
- 4 Enter an identifier string in the text field, and click OK.

Apply a bitmap as a fill

To apply a bitmap as a fill to a graphic object, use the Color panel. Applying a bitmap as a fill tiles the bitmap to fill the object. The Gradient Transform tool allows you to scale, rotate, or skew an image and its bitmap fill.

- 1 To apply the fill to existing artwork, select a graphic object or objects on the Stage.
- 2 Select Window > Color.
- 3 Select Bitmap from the pop-up menu in the upper right of the panel.
- 4 To use a larger preview window to display more bitmaps in the current document, click the arrow in the lower-right corner to expand the Color panel.
- 5 Click a bitmap to select it.

The bitmap becomes the current fill color. If you selected artwork in step 1, the bitmap is applied as a fill to the artwork.

See also

[“Transform gradient and bitmap fills”](#) on page 134

Edit a bitmap in an external editor

If you are editing a Fireworks PNG file imported as a flattened image, edit the PNG source file for the bitmap, when available.

Note: You cannot edit bitmaps from Fireworks PNG files imported as editable objects in an external image editor.

If you have Fireworks 3 or later or another image-editing application installed on your system, you can start the application from Flash to edit an imported bitmap.

Edit a bitmap with Fireworks 3 or later

- 1 In the Library panel, right-click (Windows) or Control-click (Macintosh) the bitmap's icon and select Edit With Fireworks 3.
- 2 Specify whether to open the PNG source file or the bitmap file.
- 3 Perform the desired modifications to the file in Fireworks.
- 4 In Fireworks, select File > Update.
- 5 Return to Flash.

The file is automatically updated in Flash.

Edit a bitmap with another image-editing application

- 1 In the Library panel, right-click (Windows) or Control-click (Macintosh) the bitmap's icon, and select Edit With.
- 2 Select an image-editing application to open the bitmap file, and click OK.
- 3 Perform the desired modifications to the file in the image-editing application.

- 4 Save the file in the image-editing application.

The file is automatically updated in Flash.

- 5 Return to Flash to continue editing the document.

Break apart a bitmap and create a bitmap fill

Breaking apart a bitmap on the Stage separates the on-Stage image from its library item and converts it from a bitmap instance to a shape. When you break apart a bitmap, you can modify the bitmap with the Flash drawing and painting tools. Using the Lasso tool with the Magic Wand modifier, you can select areas of the bitmap that contain the same or similar colors.

To paint with a broken-apart bitmap, select the bitmap with the Eyedropper tool and apply the bitmap as a fill with the Paint Bucket tool or another drawing tool.

See also

[“Adjust Stroke and Fill color”](#) on page 130

Break a bitmap apart

- 1 Select a bitmap in the current scene.
- 2 Select Modify > Break Apart.

Change the fill of areas of a broken-apart bitmap

- 1 Select the Lasso tool, click the Magic Wand modifier, and set the following options:
 - For Threshold, enter a value between 1 and 200 to define how closely the color of adjacent pixels must match to be included in the selection. A higher number includes a broader range of colors. If you enter 0, only pixels of the exact same color as the first pixel you click are selected.
 - For Smoothing, select an option to define how much the edges of the selection are smoothed.
- 2 To select an area, click the bitmap. To add to the selection, continue clicking.
- 3 To fill the selected areas in the bitmap, select the fill to use from the Fill Color control.
- 4 To apply the new fill, select the Paint Bucket tool and click anywhere in the selected area.

Use the Eyedropper tool to apply a bitmap fill

- 1 Select the Eyedropper tool and click the broken-apart bitmap on the Stage. The Eyedropper tool sets the bitmap to be the current fill and changes the active tool to the Paint Bucket.
- 2 Do one of the following:
 - To apply the bitmap as a fill, click an existing graphic object with the Paint Bucket tool.
 - Select the Oval, Rectangle, or Pen tool, and draw a new object. The object is filled with the broken-apart bitmap.

To scale, rotate, or skew the bitmap fill, use the Paint Bucket tool.

Convert a bitmap to a vector graphic

The Trace Bitmap command converts a bitmap into a vector graphic with editable, discrete areas of color. You manipulate the image as a vector graphic, and you can reduce the file size.

When you convert a bitmap to a vector graphic, the vector graphic is no longer linked to the bitmap symbol in the Library panel.

***Note:** If the imported bitmap contains complex shapes and many colors, the converted vector graphic might have a larger file size than the original bitmap. To find a balance between file size and image quality, try a variety of settings in the Trace Bitmap dialog box.*

You can also break apart a bitmap to modify the image with Flash drawing and painting tools.

- 1 Select a bitmap in the current scene.
- 2 Select Modify > Bitmap > Trace Bitmap.
- 3 Enter a Color Threshold value.

When two pixels are compared, if the difference in the RGB color values is less than the color threshold, the two pixels are considered the same color. As you increase the threshold value, you decrease the number of colors.

- 4 For Minimum Area, enter a value to set the number of surrounding pixels to consider when assigning a color to a pixel.
- 5 For Curve Fit, select an option to determine how smoothly outlines are drawn.
- 6 For Corner Threshold, select an option to determine whether sharp edges are retained or smoothed out.

To create a vector graphic that looks most like the original bitmap, enter the following values:

- Color Threshold: 10
- Minimum Area: 1 pixel
- Curve Fit: Pixels
- Corner Threshold: Many Corners

Chapter 5: Creating and Editing Artwork

Drawing

About drawing

The drawing tools in Adobe® Flash® CS4 Professional let you create and modify shapes for the artwork in your documents.

Before you draw and paint in Flash, it is important to understand how Flash creates artwork, and how drawing, painting, and modifying shapes can affect other shapes on the same layer.

For a text tutorial on drawing, see “Draw in Flash” on the Flash Tutorials page at www.adobe.com/go/learn_fl_tutorials.

For a video tutorial about drawing, see www.adobe.com/go/vid0119. This video describes drawing in Flash CS3 and later.

Vector and bitmap graphics

Computers display graphics in either vector or bitmap format. Understanding the difference between the two formats helps you work more efficiently. Using Flash, you can create and animate compact vector graphics. Flash also imports and manipulates vector and bitmap graphics that were created in other applications.

Vector graphics

Vector graphics describe images by using lines and curves, called *vectors*, that also include color and position properties. For example, the image of a leaf is described by points through which lines pass, creating the leaf’s outline. The color of the leaf is determined by the color of the outline and the color of the area enclosed by the outline.



Lines in vector art.

When you edit a vector graphic, you modify the properties of the lines and curves that describe its shape. Move, resize, reshape, and change the color of a vector graphic without changing the quality of its appearance. Vector graphics are resolution independent; that is, they can be displayed on output devices of varying resolutions without losing any quality.

Bitmap graphics

Bitmap graphics describe images using colored dots, called *pixels*, arranged in a grid. For example, the image of a leaf is described by the specific location and color value of each pixel in the grid, creating an image in much the same manner as a mosaic.



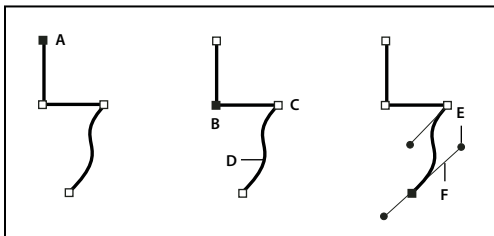
Pixels in bitmap art.

When you edit a bitmap graphic, you modify pixels rather than lines and curves. Bitmap graphics are resolution dependent, because the data describing the image is fixed to a grid of a particular size. Editing a bitmap graphic can change the quality of its appearance. In particular, resizing a bitmap graphic can make the edges of the image ragged as pixels are redistributed within the grid. Displaying a bitmap graphic on an output device that has a lower resolution than the image itself also degrades its quality.

Paths

Whenever you draw a line or shape in Flash, you create a line called a *path*. A path is made up of one or more straight or curved *segments*. The beginning and end of each segment is denoted by *anchor points*, which work like pins holding a wire in place. A path can be *closed* (for example, a circle), or *open*, with distinct *endpoints* (for example, a wavy line).

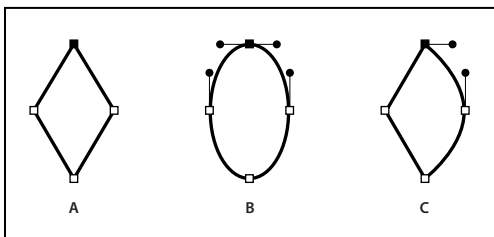
You change the shape of a path by dragging its anchor points, the *direction points* at the end of *direction lines* that appear at anchor points, or the path segment itself.



Components of a path

A. Selected (solid) endpoint B. Selected anchor point C. Unselected anchor point D. Curved path segment E. Direction point F. Direction line.

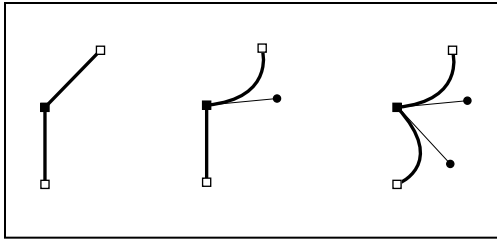
Paths can have two kinds of anchor points: corner points and smooth points. At a *corner point*, a path abruptly changes direction. At a *smooth point*, path segments are connected as a continuous curve. You can draw a path using any combination of corner and smooth points. If you draw the wrong type of point, you can always change it.



Points on a path

A. Four corner points B. Four smooth points C. Combination of corner and smooth points.

A corner point can connect any two straight or curved segments, while a smooth point always connects two curved segments.



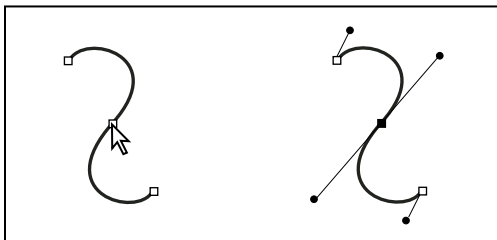
A corner point can connect both straight segments and curved segments.

Note: Don't confuse corner and smooth points with straight and curved segments.

A path outline is called a *stroke*. A color or gradient applied to an open or closed path interior area is called a *fill*. A stroke can have weight (thickness), color, and a dash pattern. After you create a path or shape, you can change the characteristics of its stroke and fill.

Direction lines and direction points

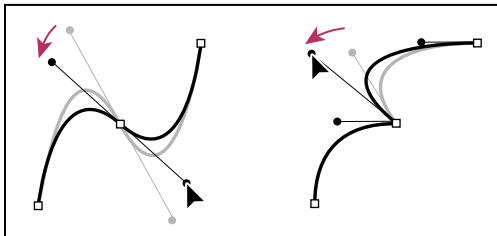
When you select an anchor point that connects curved segments (or select the segment itself), the anchor points of the connecting segments display *direction handles*, which consist of *direction lines* that end in *direction points*. The angle and length of the direction lines determine the shape and size of the curved segments. Moving the direction points reshapes the curves. Direction lines don't appear in the final output.



After selecting an anchor point (left), direction lines appear on any curved segments connected by the anchor point (right).

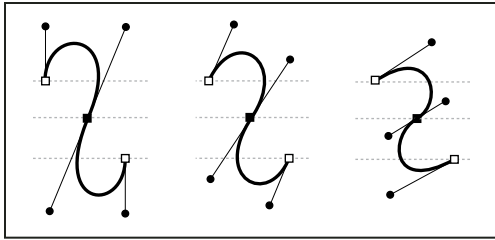
A smooth point always has two direction lines, which move together as a single, straight unit. When you move a direction line on a smooth point, the curved segments on both sides of the point are adjusted simultaneously, maintaining a continuous curve at that anchor point.

In comparison, a corner point can have two, one, or no direction lines, depending on whether it joins two, one, or no curved segments, respectively. Corner point direction lines maintain the corner by using different angles. When you move a direction line on a corner point, only the curve on the same side of the point as that direction line is adjusted.



Adjusting direction lines on a smooth point (left) and a corner point (right).

Direction lines are always tangent to (perpendicular to the radius of) the curve at the anchor points. The angle of each direction line determines the slope of the curve, and the length of each direction line determines the height, or depth, of the curve.



Moving and resizing direction lines changes the slope of curves.

Drawing modes and graphic objects

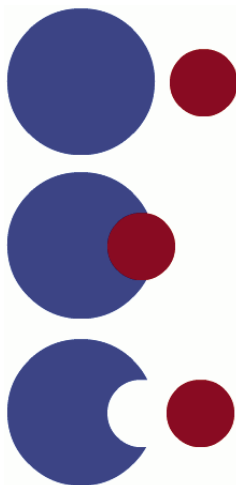
In Flash, you can create several different kinds of graphic objects by using different drawing modes and drawing tools. Each kind has its own advantages and disadvantages. By understanding the capabilities of the different graphic object types, you can make good decisions about which types of objects to use in your work.

Note: In Flash, graphic objects are items on the Stage. Flash lets you move, copy, delete, transform, stack, align, and group graphic objects. “Graphic objects” in Flash are different from “ActionScript objects,” which are part of the ActionScript® programming language. Do not confuse the two uses of the term “objects.” For more information on objects in the programming language, see *About Data Types in Learning ActionScript 2.0 in Adobe Flash*, or *Data types in Programming ActionScript 3.0*.

Merge Drawing mode

The default drawing mode automatically merges shapes that you draw when you overlap them. When you draw shapes that overlap each other in the same layer, the top-most shape cuts away the part of the shape underneath it that it overlaps. In this way, drawing shapes is a destructive drawing mode. For example, if you draw a circle and overlay a smaller circle on top of it, and then select the smaller circle and move it, the portion of the second circle that overlaid the first circle is removed.

When a shape has both a stroke and a fill, they are considered separate graphic elements that can be selected and moved independently.



Shapes created with the Merge Drawing mode merge together when they overlap. Selecting a shape and moving it results in the overlaid shape being altered.

Enter Merge Drawing mode

- 1 Select the Merge Drawing option in the Tools panel.

2 Select a drawing tool from the Tools panel and draw on the Stage

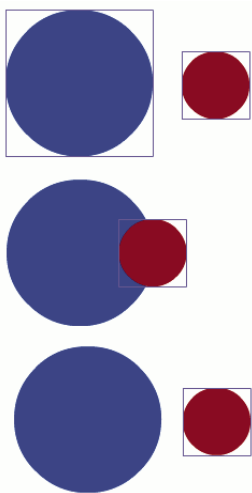
Note: By default, Flash uses the Merge Drawing mode.

Object Drawing mode

Creates shapes referred to as *drawing objects*. Drawing objects are separate graphic objects that do not automatically merge together when overlaid. This lets you overlap shapes without altering their appearance if you move them apart, or rearrange their appearance. Flash creates each shape as a separate object that you can individually manipulate.

When a drawing tool is in Object Drawing mode, the shapes you create with it are self-contained. The stroke and fill of a shape are not separate elements, and shapes that overlap do not alter one another. When you select a shape created using Object Drawing mode, Flash surrounds the shape with a rectangular bounding box to identify it.


Note: Set preferences for contact sensitivity when selecting shapes created using the Object Drawing mode.



Shapes created with the Object Drawing mode remain separate objects that you can manipulate individually.

Enter Object Drawing mode

To draw shapes using Object Drawing mode, you must explicitly enable it.

- 1 Select a drawing tool that supports Object Drawing mode (the Pencil, Line, Pen, Brush, Oval, Rectangle, and Polygon tools).
- 2 Select the Object Drawing button  from the Options category of the Tools panel, or press the J key to toggle between the Merge and Object Drawing modes. The Object Drawing button toggles between the Merge and Object Drawing modes. You can set preferences for contact sensitivity when you select shapes created with Object Drawing mode.
- 3 Draw on the Stage.

Convert a Merge Drawing mode created shape to an Object Drawing mode shape



- 1 Select the shape on the Stage.
- 2 To convert the shape into an Object Drawing mode shape, select Modify > Combine Objects > Union. After conversion, the shape is treated like a vector-based drawing object that does not alter its appearance by interacting with other shapes.

Note: To join two or more shapes into a single, object-based shape, use the Union command.

Primitive objects

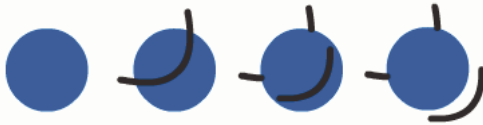
Primitive objects are shapes that let you adjust their characteristics in the Property inspector. You can precisely control the size, corner radius, and other properties of the shape at any time after you have created it without having to draw it from scratch.

Two types of primitives are available, rectangles and ovals.

- 1 Select the Rectangle Primitive Tool  or Oval Primitive Tool  from the Tools panel.
- 2 Draw on the Stage.

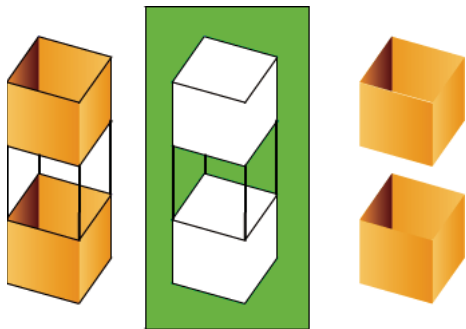
Overlapping shapes

When you draw a line across another line or painted shape in Merge Drawing mode, the overlapping lines are divided into segments at the intersection points. To select, move, and reshape each segment individually, use the Selection tool.



A fill; the fill with a line drawn through it; and the three line segments created by segmentation.

When you paint on top of shapes and lines, the portion underneath is replaced by whatever is on top. Paint of the same color merges together. Paint of different colors remains distinct. To create masks, cutouts, and other negative images, use these features. For example, the following cutout is made by moving the ungrouped kite image onto the green shape, deselecting the kite, and then moving the filled portions of the kite away from the green shape.



Making a cutout with the kite image.

To avoid inadvertently altering shapes and lines by overlapping them, group the shapes or use layers to separate them.

See also

[“Group objects”](#) on page 121

[“Create and organize layers”](#) on page 173

Drawing preferences

Set drawing settings to specify snapping, smoothing, and straightening behaviors. Change the tolerance setting for each option, and turn each option off or on. Tolerance settings are relative, depending on the resolution of your computer screen and the current magnification of the scene. By default, each option is on and set to Normal tolerance.

Drawing settings

- 1 Select Edit > Preferences (Windows) or Flash > Preferences (Macintosh) and select Drawing.
- 2 Under the Drawing category, select from the following options:

Pen tool Lets you set options for the Pen tool. Select Show Pen Preview to display a preview line from the last clicked point to the current location of the pointer. Select Show Solid Points to display control points as small filled squares instead of unfilled squares. Select Show Precise Cursors to display a crosshair cursor instead of the Pen tool icon when using the Pen tool. This option lets you see the exact target of clicks more easily.

Connect Lines Determines how close the end of a line being drawn must be to an existing line segment before the end point snaps to the nearest point on the other line. This setting also controls horizontal and vertical line recognition—that is, how nearly horizontal or vertical a line must be drawn before Flash makes it exactly horizontal or vertical. When Snap To Objects is turned on, this setting controls how close objects must be to snap to one another.

Smooth Curves Specifies the amount of smoothing applied to curved lines drawn with the Pencil tool when the drawing mode is set to Straighten or Smooth. (Smoother curves are easier to reshape, whereas rougher curves match the original line strokes more closely.)

Note: To further smooth existing curved segments, use Modify > Shape > Smooth and Modify > Shape > Optimize.

Recognize Lines Defines how nearly straight a line segment drawn with the Pencil tool must be before Flash recognizes it as a straight line and makes it perfectly straight. If Recognize Lines is off while you draw, straighten lines later by selecting one or more line segments and selecting Modify > Shape > Straighten.

Recognize Shapes Controls how precisely to draw circles, ovals, squares, rectangles, and 90° and 180° arcs for them to be recognized as geometric shapes and redrawn accurately. The options are Off, Strict, Normal, and Tolerant. Strict requires that the shape be drawn very close to straight; Tolerant specifies that the shape can be somewhat rough, and Flash redraws the shape. If Recognize Shapes is off while you draw, straighten lines later by selecting one or more shapes (for example, connected line segments) and selecting Modify > Shape > Straighten.

Click Accuracy Specifies how close to an item the pointer must be before Flash recognizes the item.

Specify the Selection, Subselection, and Lasso tool contact-sensitivity options when you create shapes using Object Drawing mode. By default, objects are only selected when the tool's marquee rectangle completely surrounds the object. Deselecting this option selects entire objects when they are only partially enclosed by the selection marquee of the Selection, Subselection, or Lasso tools.

Selection, Subselection, and Lasso tool contact options


- 1 Select Edit > Preferences (Windows) or Flash > Preferences (Macintosh).
- 2 In the General category, do one of the following:
 - To select only objects and points that are completely enclosed by the selection marquee, deselect Contact-Sensitive Selection and Lasso tools. Points that lie within the selection area are still selected.
 - To select objects or groups that are only partially enclosed by the selection marquee, select Contact-Sensitive Selection and Lasso tools.

Note: The Subselection tools use the same contact-sensitive setting.


Draw simple lines and shapes

Draw straight lines with the Line Segment tool

To draw one straight line segment at a time, use the Line tool.

- 1 Select the Line tool .
- 2 Select Window > Properties and select stroke attributes.

Note: You cannot set fill attributes for the Line tool.

- 3 Click the Object Drawing button  in the Options section of the Tools panel to select either Merge or Object Drawing mode. When the Object Drawing button is depressed, the Line tool is in Object Drawing mode.
- 4 Position the pointer where the line is to begin, and drag to where the line is to end. To constrain the angle of the line to multiples of 45°, Shift-drag.

See also

[“Adjust Stroke and Fill color”](#) on page 130

[“Drawing modes and graphic objects”](#) on page 94

Draw rectangles and ovals

The Oval and Rectangle tools let you create these basic geometric shapes, and apply strokes, fills, and specify rounded corners. In addition to the Merge and Object drawing modes, the Oval and Rectangle tools also provide the Primitive Object drawing mode.



When you create rectangles or ovals using the Rectangle Primitive or Oval Primitive tools, Flash draws the shapes as separate objects not unlike the shapes you create using Object Drawing mode. The primitive shape tools let you specify the corner radius of rectangles, and the start and end angle and the inner radius of ovals using controls in the Property inspector. After you create a primitive shape, alter the radius and dimensions by selecting the shape on the Stage and adjusting the controls in the Property inspector.

Note: When either of the Primitive Object drawing tools is selected, the Property inspector retains the values of the last primitive object that you edited. For example, if you modify a rectangle and then draw a second rectangle.

See also

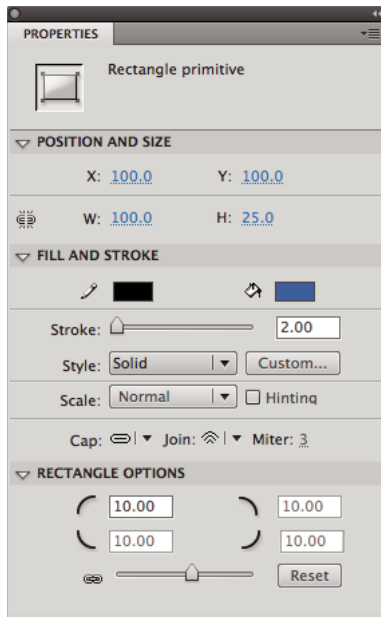
[“Adjust Stroke and Fill color”](#) on page 130

Draw rectangle primitives

- 1 To select the Rectangle Primitive tool, click and hold the mouse button on the Rectangle tool , and select the Rectangle Primitive tool  from the pop-up menu.
- 2 To create a rectangle primitive, drag with the Rectangle Primitive tool on the Stage.

Note: To change the corner radius while dragging with the Rectangle primitive tool, press the Up Arrow key or Down Arrow key. When the corners achieve the desired roundness, release the key.

- 3 With the rectangle primitive selected, you can use the controls in the Property inspector to further modify the shape or specify fill and stroke colors.



Properties for a rectangle primitive.



These Property inspector controls are specific to the Rectangle Primitive tool:

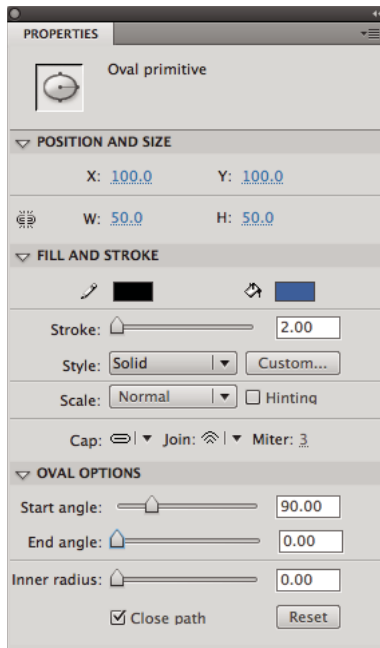
Rectangle Corner Radius Controls Let you specify the corner radii for the rectangle. You can enter a numeric value for the inner radius in each text box. Entering a negative value creates an inverse radius. You can also deselect the constrain corner radius icon, and adjust each corner radius individually.

Reset Resets all of the Rectangle Primitive tool controls, and restores the rectangle primitive shape drawn on the Stage to its initial size and shape.

- 4 To specify a different corner radius for each corner, deselect the Lock icon in the Rectangle Options area of the Property inspector. When locked, the radius controls are restrained so that each corner uses the same radius.
- 5 To reset the corner radii, click the Reset button in the Property inspector.

Draw oval primitives

- 1 Click and hold the mouse button on the Rectangle tool , and select the Oval Primitive tool .
- 2 To create an oval primitive, drag the Primitive Oval tool on the Stage. To constrain the shape to a circle, Shift-drag.
- 3 With the oval primitive selected on the Stage, you can use the controls found in the Property inspector to further modify the shape or specify fill and stroke colors.



Properties for an oval primitive.

These Property inspector controls are specific to the Oval Primitive tool:

Start Angle/End Angle The angle of the start point and end point of the oval. Using these controls, you can easily modify the shape of ovals and circles into pie slices, half circles, and other creative shapes.



Inner Radius An inner radius (or oval) within the oval. You can either enter a numeric value for the inner radius in the box or click the slider and interactively adjust the size of the inner radius. You can enter values from 0 to 99 representing the percentage of fill that is removed.

Close Path Determines whether the path (or paths, if you are specifying an inner radius) of the oval is closed. If you specify an open path, no fill is applied to the resulting shape, only the stroke is drawn. Close Path is selected by default.

Reset Resets all of the Oval Primitive tool controls and restores the oval primitive shape drawn on the Stage to its initial size and shape.

Draw ovals and rectangles

The Oval and Rectangle tools create these basic geometric shapes.


- 1 To select the Rectangle tool  or Oval tool , click and hold the mouse button on the Rectangle tool and drag.
- 2 To create a rectangle or oval, drag the Rectangle or Oval tool on the Stage.
- 3 For the Rectangle tool, specify rounded corners by clicking the Round Rectangle modifier and entering a corner radius value. A value of zero (0) creates square corners.
- 4 Drag on the Stage. If you are using the Rectangle tool, press the Up Arrow and Down Arrow keys while dragging to adjust the radius of rounded corners.

For the Oval and Rectangle tools, Shift-drag to constrain the shapes to circles and squares.

- 5 To specify a specific size of oval or rectangle, select the Oval or Rectangle tool and press the Alt key (Windows) or Option key (Macintosh). Then click the Stage to display the Oval And Rectangle Settings dialog box.
 - For ovals, specify the width and height in pixels and whether to draw the oval from the center.





- For rectangles, specify the width and height in pixels, the radius of the rounded corners, and whether to draw the rectangle from the center.

Draw polygons and stars

- 1 Select the PolyStar tool  by clicking and holding the mouse button on the Rectangle tool and selecting from the pop-up menu that appears.
- 2 Select Window > Properties and select fill and stroke attributes.
- 3 Click Options and do the following:
 - For Style, select Polygon or Star.
 - For Number Of Sides, enter a number from 3 through 32.
 - For Star Point Size, enter a number from 0 through 1 to specify the depth of the star points. A number closer to 0 creates deeper points (like needles). If you are drawing a polygon, leave this setting unchanged. (It does not affect the polygon shape.)
- 4 Click OK.
- 5 Drag on the Stage.

Draw with the Pencil tool

To draw lines and shapes, use the Pencil tool, in much the same way that you use a real pencil to draw. To apply smoothing or straightening to the lines and shapes as you draw, select a drawing mode for the Pencil tool.


- 1 Select the Pencil tool .
- 2 Select Window > Properties and select a stroke color, line weight, and style.
- 3 Select a drawing mode under Options in the Tools panel:
 - To draw straight lines and convert approximations of triangles, ovals, circles, rectangles, and squares into these common geometric shapes, select Straighten .
 - To draw smooth curved lines, select Smooth .
 - To draw freehand lines with no modification applied, select Ink .



Lines drawn with Straighten, Smooth, and Ink mode, respectively.

- 4 To draw with the Pencil tool, Shift-drag to constrain lines to vertical or horizontal directions, click the Stage, and drag.

Paint with the Brush tool

The Brush tool  draws brush-like strokes. It creates special effects, including calligraphic effects. Select a brush size and shape using the Brush tool modifiers.

Brush size for new strokes remains constant even when you change the magnification level for the Stage, so the same brush size appears larger when the Stage magnification is lower. For example, suppose you set the Stage magnification to 100% and paint with the Brush tool using the smallest brush size. Then, you change the magnification to 50% and paint again using the smallest brush size. The new stroke that you paint appears 50% thicker than the earlier stroke. (Changing the magnification of the Stage does not change the size of existing brush strokes.)


Use an imported bitmap as a fill when painting with the Brush tool. See “[Break apart groups and objects](#)” on page 122.

If you have a Wacom pressure-sensitive tablet connected to your computer, vary the width and angle of the brush stroke by using the Brush tool Pressure and Tilt modifiers, and varying pressure on the stylus.

The Pressure modifier varies the width of brush strokes when you vary the pressure on the stylus. The Tilt modifier varies the angle of brush strokes when you vary the angle of the stylus on the tablet. The Tilt modifier measures the angle between the top (eraser) end of the stylus and the top (north) edge of the tablet. For example, if you hold the pen vertically against the tablet, the Tilt is 90. The Pressure and Tilt modifiers are both fully supported for the eraser function of the stylus.



A variable-width brush stroke drawn with a stylus.

- 1 Select the Brush tool .
- 2 Select Window > Properties and select a fill color.
- 3 Click the Brush Mode modifier and select a painting mode:
 - Paint Normal** Paints over lines and fills on the same layer.
 - Paint Fills** Paints fills and empty areas, leaving lines unaffected.
 - Paint Behind** Paints in blank areas of the Stage on the same layer, leaving lines and fills unaffected.
 - Paint Selection** Applies a new fill to the selection when you select a fill in the Fill Color control or the Fill box of the Property inspector, the same as selecting a filled area and applying a new fill.
 - Paint Inside** Paints the fill in which you start a brush stroke and never paints lines. If you start painting in an empty area, the fill doesn't affect any existing filled areas.
- 4 Select a brush size and brush shape from the Brush tool modifiers.
- 5 If a Wacom pressure-sensitive tablet is attached to your computer, select the Pressure modifier, the Tilt modifier, or both, to modify brush strokes.
 - Select the Pressure modifier to vary the width of your brush strokes by varying the pressure on your stylus.
 - To vary the angle of your brush strokes by varying the angle of the stylus on the Wacom pressure-sensitive tablet, select the Tilt modifier.
- 6 Drag on the Stage. To constrain brush strokes to horizontal and vertical directions, Shift-drag.

See also


“[Adjust Stroke and Fill color](#)” on page 130


Drawing with the Pen tool


To draw precise paths as straight lines or smooth, flowing curves, use the Pen tool. When you draw with the Pen tool, click to create points on straight line segments and drag to create points on curved line segments. Adjust straight and curved line segments by adjusting points on the line.


Pen tool drawing states


The Pen tool provides feedback about its current drawing state by displaying different pointers. The various drawing states are indicated by the following pointers:


Initial Anchor Point pointer  The first pointer you see when you select the Pen tool. Indicates that the next mouse click on the Stage will create an initial anchor point, which is the beginning of a new path (all new paths begin with an initial anchor point). Any existing drawing paths are terminated.


Sequential Anchor Point pointer  Indicates that the next mouse click will create an anchor point with a line connecting it to the previous anchor point. This pointer is displayed during the creation of all user-defined anchor points except the initial anchor point of a path.

Add Anchor Point pointer  Indicates that the next mouse click will add an anchor point to an existing path. To add an anchor point, the path must be selected, and the Pen tool must not be over an existing anchor point. The existing path is redrawn based on the additional anchor point. Only one anchor point can be added at a time.


Delete Anchor Point pointer  Indicates that the next mouse click on an existing path will remove an anchor point. To remove an anchor point, the path must be selected with the Selection tool, and the pointer must be over an existing anchor point. The existing path is redrawn based on the removal of the anchor point. Only one anchor point can be removed at a time.


Continue Path pointer  Extends a new path from an existing anchor point. For this pointer to be activated, the mouse must be over an existing anchor point on a path. This pointer is only available when you are not currently drawing a path. The anchor point does not have to be one of the terminal anchor points of a path; any anchor point can be the location of a continued path.

Close Path pointer  Closes the path you're drawing on the starting point of the path. You can only close a path that you are currently drawing, and the existing anchor point must be the starting anchor point of the same path. The resulting path does not have any specified fill color settings applied to the enclosed shape; apply fill color separately.

Join Paths pointer  Similar to the Close Path Tool except that the mouse must not be over the initial anchor point of the same path. The pointer must be over either of the terminal points of a unique path. The segment may or may not be selected.

Note: Joining paths may or may not result in a closed shape.

Retract Bezier Handle pointer  Appears when the mouse is over an anchor point whose Bezier handles are displayed. Clicking the mouse retracts the Bezier handles and causes the curved path across the anchor point to revert to straight segments.

Convert Anchor Point pointer  Converts a corner point without direction lines to a corner point with independent direction lines. To enable the Convert Anchor Point pointer, use the Shift + C modifier keys to toggle the Pen tool.


For a video tutorial about the Pen tool, see www.adobe.com/go/vid0120.

See also

“Reshape lines and shapes” on page 111

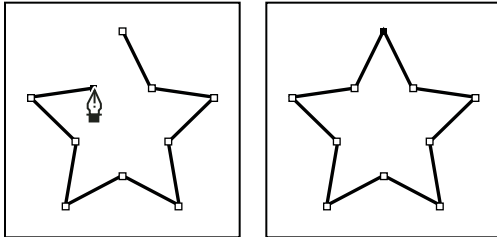
Draw straight lines with the Pen tool

The simplest path you can draw with the Pen tool is a straight line, made by clicking the Pen tool to create two anchor points. Continue to click to create a path made of straight line segments connected by corner points.


- 1 Select the Pen tool .
- 2 Position the Pen tool where the straight segment is to begin, and click to define the first anchor point. If direction lines appear, you accidentally dragged the Pen tool; choose Edit > Undo and click again.

Note: The first segment you draw is not visible until you click a second anchor point (unless you've specified Show Pen Preview in the Drawing category of the Preferences dialog box).

- 3 Click again where you want the segment to end (Shift-click to constrain the angle of the segment to a multiple of 45°).
- 4 Continue clicking to set anchor points for additional straight segments.



Clicking Pen tool creates straight segments.

- 5 To complete the path as an open or closed shape, do one of the following:
 - To complete an open path, double-click the last point, click the Pen tool in the Tools panel, or Control-click (Windows) or Command-click (Macintosh) anywhere away from the path.
 - To close the path, position the Pen tool over the first (hollow) anchor point. A small circle appears next to the Pen  tool pointer when it is positioned correctly. Click or drag to close the path.
 - To complete the shape as is, select Edit > Deselect All, or select a different tool in the Tools panel.


See also

“[Adjust Stroke and Fill color](#)” on page 130

Draw curves with the Pen tool

To create a curve, add an anchor point where a curve changes direction and drag the direction lines that shape the curve. The length and slope of the direction lines determine the shape of the curve.

Curves are easier to edit and your system can display and print them faster if you draw them using as few anchor points as possible. Using too many points can also introduce unwanted bumps in a curve. Instead, draw widely spaced anchor points, and practice shaping curves by adjusting the length and angles of the direction lines.

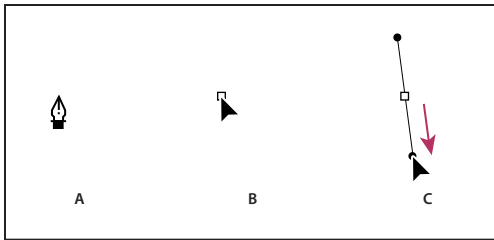
- 1 Select the Pen tool .
- 2 Position the Pen tool where the curve is to begin, and hold down the mouse button.

The first anchor point appears, and the Pen tool pointer changes to an arrowhead. (In Photoshop, the pointer changes only after you've started dragging.)

- 3 Drag to set the slope of the curve segment you're creating, and then release the mouse button.

In general, extend the direction line about one third of the distance to the next anchor point you plan to draw. (You can adjust one or both sides of the direction line later.)

Hold down the Shift key to constrain the tool to multiples of 45°.

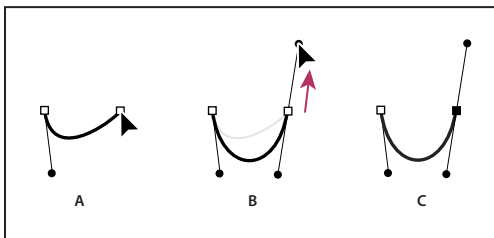


Drawing the first point in a curve

A. Positioning Pen tool B. Starting to drag (mouse button pressed) C. Dragging to extend direction lines.

4 Position the Pen tool where the curve segment is to end, and do one of the following:

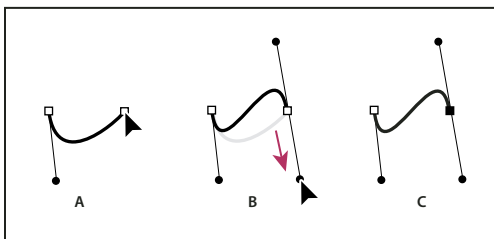
- To create a C-shaped curve, drag in a direction opposite to the previous direction line and release the mouse button.



Drawing the second point in a curve

A. Starting to drag second smooth point B. Dragging away from previous direction line, creating a C curve C. Result after releasing mouse button.


- To create an S-shaped curve, drag in the same direction as the previous direction line and release the mouse button.




Drawing an S curve

A. Starting to drag new smooth point B. Dragging in the same direction as previous direction line, creating an S curve C. Result after releasing mouse button.

5 To create a series of smooth curves, continue dragging the Pen tool from different locations. Place anchor points at the beginning and end of each curve, not at the tip of the curve.




 To break out the direction lines of an anchor point, Alt-drag (Windows) or Option-drag (Macintosh) direction lines.

6 To complete the path, do one of the following:

- To close the path, position the Pen tool over the first (hollow) anchor point. A small circle appears next to the Pen tool pointer  when it is positioned correctly. Click or drag to close the path.
- To leave the path open, Ctrl-click (Windows) or Command-click (Macintosh) anywhere away from all objects, select a different tool, or choose Edit > Deselect All.





Add or delete anchor points

Adding anchor points can give you more control over a path or it can extend an open path. However, it's a good idea not to add more points than necessary. A path with fewer points is easier to edit, display, and print. To reduce the complexity of a path, delete unnecessary points.

The toolbox contains three tools for adding or deleting points: the Pen tool , the Add Anchor Point tool , and the Delete Anchor Point tool .

By default, the Pen tool changes to the Add Anchor Point tool as you position it over a selected path, or to the Delete Anchor Point tool as you position it over an anchor point.

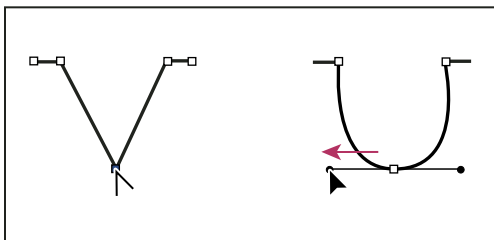
Note: Don't use the Delete, Backspace, and Clear keys or the Edit > Cut or Edit > Clear commands to delete anchor points; these keys and commands delete the point and the line segments that connect to that point.

- 1 Select the path to modify.
- 2 Click and hold the mouse button on the Pen tool , then select the Pen tool , Add Anchor Point tool , or the Delete Anchor Point tool .
- 3 To add an anchor point, position the pointer over a path segment, and click. To delete an anchor point, position the pointer over an anchor point, and click.

Adjust anchor points on paths



When you draw a curve with the Pen tool, you create smooth points—anchor points on a continuous, curved path. When you draw a straight line segment or a straight line connected to a curved segment, you create corner points—anchor points on a straight path or at the juncture of a straight and a curved path.

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



Dragging a direction point out of a corner point to create a smooth point.

Move or add anchor points

- To move an anchor point, drag the point with the Subselection tool .
- To nudge an anchor point or points, select the point or points with the Subselection tool and use the arrow keys to move the point or points. Shift-click to select multiple points.
- To add an anchor point, click a line segment with the Pen tool. A plus (+) sign appears next to the Pen tool , if an anchor point can be added to the selected line segment. If the line segment is not yet selected, click it with the Pen tool to select it, and then add an anchor point.

Delete anchor points


Deleting unneeded anchor points on a curved path optimizes the curve and reduces the resulting SWF file size.

- To delete a corner point, click the point once with the Pen tool. A minus (-) sign appears next to the Pen tool if an anchor point can be deleted from the selected line segment. If the line segment is not yet selected, click it with the Pen tool to select it, and then delete the anchor point.

- To delete a smooth point, click the point once with the Pen tool. A minus (-) sign appears next to the Pen tool if an anchor point can be deleted from the selected line segment. If the line segment is not yet selected, click it with the Pen tool to select it, and then delete the corner point. (Click once to convert the point to a corner point, and once more to delete the point.)

Convert segments between straight and curved


To convert segments in a line from straight segments to curve segments, convert corner points to smooth points. You can also do the reverse.

- To convert a corner point to a smooth point, use the Subselection tool to select the point, then Alt-drag (Windows) or Option-drag (Macintosh) the point to place the tangent handles.
- To convert a smooth point to a corner point, click the point with the Pen tool. The carat ^ marker next to the pointer  indicates when it is over the smooth point.

Adjust segments

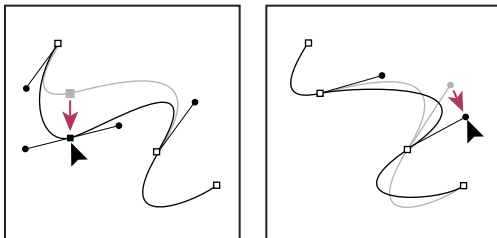
To change the angle or length of the segment or adjust curved segments to change the slope or direction of the curve, adjust straight segments.

When you move a tangent handle on a smooth point, the curves on both sides of the point adjust. When you move a tangent handle on a corner point, only the curve on the same side of the point as the tangent handle adjusts.

- To adjust a straight segment, select the Subselection tool , and select a straight segment. Use the Subselection tool to drag an anchor point on the segment to a new position.
- To adjust a curve segment, select the Subselection tool and drag the segment.

***Note:** When you click the path, Flash shows the anchor points. Adjusting a segment with the Subselection tool can add points to the path.*


- To adjust points or tangent handles on a curve, select the Subselection tool, and select an anchor point on a curved segment.
- To adjust the shape of the curve on either side of the anchor point, drag the anchor point, or drag the tangent handle. To constrain the curve to multiples of 45°, Shift-drag. To drag tangent handles individually, Alt-drag (Windows) or Option-drag (Macintosh).



Drag the anchor point, or drag the direction point.

Pen tool preferences

Specify preferences for the appearance of the Pen tool pointer, for previewing line segments as you draw, and for the appearance of selected anchor points. Selected line segments and anchor points use the outline color of the layer on which the lines and points appear.

- 1 Select the Pen tool , then select Edit > Preferences (Windows) or Flash > Preferences (Macintosh).
- 2 In the Category list, select Drawing.

3 Set the following options for the Pen tool:

Show Pen Preview Previews line segments as you draw. A preview of the line segment appears as you move the pointer around the Stage, before you click to create the end point of the segment. If this option is not selected, a line segment does not appear until you create the end point.

Show Solid Points Displays selected anchor points as hollow and deselected anchor points as solid. If this option is not chosen, selected anchor points are solid, and deselected anchor points are hollow.

Show Precise Cursors Specifies that the Pen tool pointer appears as a cross-hair pointer rather than the default Pen tool icon, for more precise placement of lines. To display the default Pen tool icon with the Pen tool, deselect the option.

Note: To switch between the cross-hair pointer and the default Pen tool icon, press **Caps Lock**.

4 Click OK.

Drawing patterns with the Decorative drawing tools

The Decorative drawing tools let you turn graphic shapes that you create into complex, geometric patterns. The Decorative drawing tools use algorithmic calculations—known as *procedural drawing*. These calculations are applied to a movie clip or graphic symbol in the library that you create. In this way, you can create a complex pattern using any graphic shape or object. Use the Spray Brush or Fill tool to apply the patterns that you create. You can create kaleidoscopic effects using single or multiple symbols with the Deco Symmetry tool.

For a video tutorial about the Decorative drawing tools, see Using the Decorative Drawing Tools at www.adobe.com/go/lrvid4060_fl.

Jonathan Duran has provided an in-depth article entitled Using the Deco tool and Spray Brush for creating complex, geometric patterns in Flash at www.adobe.com/devnet/flash/articles/deco_intro.html.

Apply patterns with the Spray Brush tool

The Spray Brush acts like a particle sprayer, letting you “brush” a pattern of shapes onto the Stage at one time. By default, the Spray Brush emits a spray of dot particles using the currently selected fill color. However, you can use the Spray Brush tool to apply a movie clip or graphic symbol as a pattern.

- 1 Select the Spray Brush tool.
- 2 In the Spray Brush tool Property inspector, select a fill color for the default spray of dots. Or, click Edit to select a custom symbol from the library.

You can use any movie clip or graphic symbol in the library as a “particle.” These symbol-based particles give you a great deal of creative control over artwork you create in Flash.

- 3 Click or drag on the Stage where you want the pattern to appear.

See also

“Create symbols” on page 148

“Working with the library” on page 157

Spray Brush tool options

The Spray Brush tool options appear in the Property inspector when you select the Spray Brush from the Tools panel.

Edit Opens the Select Symbol dialog box, from which you select a movie clip or graphic symbol to use as a spray brush particle. When a symbol in the library is selected, its name appears next to the edit button.

Color selector Select a fill color for the default particle spray. The color selector is disabled when you use a symbol in the library as a spray particle.

Scale width Scale the width of a symbol used as a spray particle. For example, a value of 10% makes the symbol 10% narrower. A value of 200% makes the symbol 200% wider.

Scale height Scales the height of a symbol used as a spray particle. For example, a value of 10% makes the symbol 10% shorter. A value of 200% makes the symbol 200% taller.

Random scaling Specifies that each symbol-based spray particle is placed on Stage at a random scale, altering the size of each particle. This option is disabled when using the default spray of dots.

Rotate symbol Rotates the symbol-based spray particle around a center point.

Random rotation Specifies that each symbol-based spray particle is placed on Stage at a random degree of rotation. This option is disabled when you use the default spray of dots.

The Deco Drawing tool

Use the Deco Drawing tool to apply an effect to a selected object on the Stage. Choose the effects from the Property inspector after you select the Deco Drawing tool.

Apply the Symmetry effect

Use the Symmetry effect to arrange symbols symmetrically around a central point. When you draw the symbols on the Stage, a set of handles appears. Use the handles to control the symmetry by increasing the number of symbols, adding symmetries, or editing and modifying the effect.

Use the Symmetry effect to create circular user interface elements (such as an analog clock face or dial gauge) and swirl patterns. The default symbol for the Symmetry effect is a black rectangle shape with no stroke that is 25 x 25 pixels.

- 1 Select the Deco Drawing tool, and select Symmetry Brush from the Drawing Effect menu in the Property inspector.
- 2 In the Deco Drawing tool Property inspector, select a fill color to use for the default rectangle shape. Or, click Edit to select a custom symbol from the library.

You can use any movie clip or graphic symbol in the library with the Symmetry Brush effect. These symbol-based particles give you a great deal of creative control over artwork you create in Flash.

- 3 The Symmetry Brush advanced options appear in the Property inspector when you select Symmetry Brush from the Drawing Effect pop-up menu in the Property inspector.

Rotate Around Point Rotates the shapes in the symmetry around a fixed point that you designate. The default reference point is the center point of the symmetry. To rotate the object around its center point, drag in a circular motion.

Reflect Across Line Flips the shapes an equal distance apart across an invisible line that you specify.

Reflect Around Point Places two shapes an equal distance apart around a fixed point that you specify.

Grid Translation Creates a grid using the shapes in the Symmetry effect you are drawing. Each click of the Deco Drawing tool on the Stage creates a grid of shapes. Adjust the height and width of the shapes using the x and y coordinates defined by the Symmetry Brush handles.

Test Collisions Prevents shapes in the Symmetry effect you are drawing from colliding with one another, regardless of how you increase the number of instances within the Symmetry effect. Deselect this option to overlap the shapes in the Symmetry effect.

- 4 Click the Stage where you want the Symmetry Brush artwork to appear.
- 5 Use the Symmetry Brush handles to adjust the size of the symmetry and the number of symbol instances.

Apply the Grid Fill effect

The Grid Fill effect lets you fill the Stage, a symbol, or closed region with a symbol from the library. After the Grid Fill is drawn to the Stage, if the filled symbol is moved or resized, the Grid Fill will move or resize accordingly.

Use the Grid Fill effect to create a checkerboard, a tiled background, or area or shape with a custom pattern. The default symbol for the Symmetry effect is a black rectangle shape with no stroke that is 25 x 25 pixels.

- 1 Select the Deco Drawing tool, and select Grid Fill from the Drawing Effect menu in the Property inspector.
- 2 In the Deco Drawing tool Property inspector, select a fill color for the default rectangle shape. Or click Edit to select a custom symbol from the library.

You can use any movie clip or graphic symbol in the library as a symbol with the Grid Fill effect.

- 3 You can specify the horizontal and vertical spacing and the scale of the fill shape. Once the Grid Fill effect is applied, you cannot change the advanced options in the Property inspector afterwards to alter the fill pattern.

Horizontal spacing Specifies the horizontal distance in pixels between shapes used in the Grid Fill.

Vertical spacing Specifies the vertical distance in pixels between shapes used in the Grid Fill.

Pattern scale Enlarges or shrinks an object horizontally (along the x axis), and vertically (along the y axis).

- 4 Click the Stage or within the shape or symbol where you want the Grid Fill pattern to appear.

Apply the Vine Fill effect

The Vine Fill effect lets you fill the Stage, a symbol, or closed region with a vine pattern. You can substitute your own artwork for the leaves and flowers by selecting symbols from the library. The resulting pattern is contained in a movie clip that itself contains the symbols that make up the pattern.

- 1 Select the Deco Drawing tool, and select Vine Fill from the Drawing Effect menu in the Property inspector.
- 2 In the Deco Drawing tool Property inspector, select a fill color for the default flower and leaf shapes. Or, click Edit to select a custom symbol from the library to replace one or both of the default flower and leaf symbols.

You can use any movie clip or graphic symbol in the library to replace the default flower and leaf symbols with the Vine Fill effect.

- 3 You can specify the horizontal and vertical spacing and the scale of the fill shape. After you apply the Vine Fill effect, you cannot change the advanced options in the Property inspector afterward to alter the fill pattern.

Branch angle Specifies the angle of the branch pattern.

Branch color Specifies the color to use for the branch.

Pattern scale Scaling an object enlarges or reduces it both horizontally (along the x axis), and vertically (along the y axis).

Segment length Specifies the length of the segments between leaf and flower nodes.

Animate pattern Specifies that each iteration of the effect is drawn to a new frame in the timeline. This option creates a frame-by-frame animated sequence of the flower pattern as it's drawn.


Frame step Specifies how many frames to span per second of the effect being drawn.

- 4 Click the Stage or within the shape or symbol where you want the Grid Fill pattern to appear.

Reshaping objects

Reshape lines and shapes

Display and adjust points with the Subselection tool

- 1 Select the Subselection tool .
- 2 Click the line or shape outline.

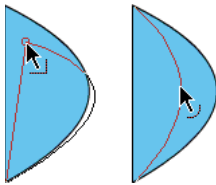
See also

“[Adjust anchor points on paths](#)” on page 106

Reshape a line or shape

To reshape a line or shape outline, drag any point on a line using the Selection tool. The pointer changes to indicate what type of reshaping it can perform on the line or fill.


Flash adjusts the curve of the line segment to accommodate the new position of the moved point. If the repositioned point is an end point the line will lengthen or shorten. If the repositioned point is a corner, the line segments forming the corner remain straight as they become longer or shorter.



When a corner appears next to the pointer, you can change an end point. When a curve appears next to the pointer, you can adjust a curve.

Some brush stroke areas are easier to reshape if you view them as outlines.

If you are having trouble reshaping a complex line, smooth it to remove some of its details, making reshaping easier. Increasing the magnification can also make reshaping easier and more accurate.

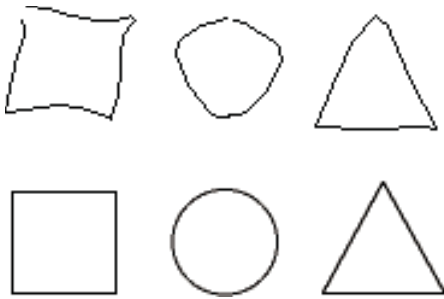
- 1 Select the Selection tool .
- 2 Do one of the following:
 - To reshape the segment, drag from any point.
 - To drag a line to create a new corner point, control-click (Windows) or Option-click (Macintosh).

Straighten and smooth lines

Straightening makes small straightening adjustments to lines and curves you already drew. It has no effect on segments that are already straight.

Note: To adjust the degree of automatic smoothing and straightening, specify preferences for drawing settings.






To make Flash recognize shapes, use the straightening technique. If you draw any oval, rectangular, or triangular shapes with the Recognize Shapes option turned off, use the Straightening option to make the shapes geometrically perfect. Shapes that are touching, and thus connected to other elements, are not recognized.



Shape recognition turns the top shapes into the bottom shapes.

Smoothing softens curves and reduces bumps or other variations in a curve's overall direction. It also reduces the number of segments in a curve. Smoothing is relative, however, and has no effect on straight segments. It is particularly useful when you are having trouble reshaping a number of very short curved line segments. Selecting all the segments and smoothing them reduces the number of segments, producing a gentler curve that is easier to reshape.

Repeated application of smoothing or straightening makes each segment smoother or straighter, depending on how curved or straight each segment was originally.

- To smooth the curve of each selected stroke, select the Selection tool and click the Smooth modifier  in the Options section of the Tools panel. Each click of the Smooth modifier button makes the selected stroke progressively smoother.
- To enter specific parameters for a smoothing operation, choose Modify > Shape > Smooth. In the Smooth dialog box, enter values for the Smooth Angles Above, Smooth Angles Below, and Smoothing Strength parameters.
- To make small straightening adjustments on each selected fill outline or curved line, select the Selection tool  and click the Straighten modifier  in the Options section of the Tools panel.
- To enter specific parameters for a straightening operation, choose Modify > Shape > Straighten. In the Straighten dialog box, enter a value for the Straighten Strength parameter.
- To use shape recognition, select the Selection tool  and click the Straighten modifier , or select Modify > Shape > Straighten.

See also

["Drawing preferences"](#) on page 96

Optimize curves

Optimizing smooths curves by refining curved lines and filling outlines, reducing the number of curves used to define these elements. Optimizing curves also reduces the size of the Flash document (FLA file) and the exported Flash application (SWF file). Apply optimization to the same elements multiple times.

- 1 Select the drawn elements to optimize and select Modify > Shape > Optimize.
- 2 To specify the degree of smoothing, drag the Optimization Strength slider. The results depend on the curves selected. Generally, optimizing produces fewer curves, with less resemblance to the original outline.
- 3 To display a message indicating the number of segments in the selection before and after the optimization, select the Show Totals Message option. Flash displays the message after the operation completes.
- 4 Click OK.

Modify shapes

- 1 To convert lines to fills, select a line or multiple lines and select **Modify > Shape > Convert Lines To Fills**. Selected lines are converted to filled shapes, which allows you to fill lines with gradients or to erase a portion of a line. Converting lines to fills can make file sizes larger, but it can also speed up drawing for some animations.
- 2 To expand the shape of a filled object, select a filled shape, and select **Modify > Shape > Expand Fill**. Enter a value in pixels for **Distance** and select **Expand** or **Inset For Direction**. **Expand** enlarges the shape, and **Inset** reduces it. This feature works best on a single, small, filled color shape with no stroke, that does not contain many small details.
- 3 To soften the edges of an object, select a filled shape, and select **Modify > Shape > Soften Fill Edges**. Set the following options:


Distance The width, in pixels, of the soft edge.

Number Of Steps Controls how many curves are used for the soft edge effect. The more steps you use, the smoother the effect. Increasing steps also creates larger files and slows drawing.


Expand Or Inset Controls whether the shape is enlarged or reduced to soften the edges.

This feature works best on a single filled shape that has no stroke, and can increase the file size of a Flash document and the resulting SWF file.

Delete everything on the Stage

- ❖ Double-click the Eraser tool  in the toolbar. This erases all types of content on the Stage and pasteboard.

Remove stroke segments or filled areas

- 1 Select the Eraser tool, and then click the Faucet modifier .
- 2 Click the stroke segment or filled area to delete.

Erase by dragging

- 1 Select the Eraser tool.
- 2 Click the Eraser Mode modifier and select an erasing mode:

Erase Normal Erases strokes and fills on the same layer.

Erase Fills Erases only fills; strokes are not affected.

Erase Lines Erases only strokes; fills are not affected.

Erase Selected Fills Erases only the currently selected fills and does not affect strokes, selected or not. (Select the fills to erase before using the Eraser tool in this mode.)

Erase Inside Erases only the fill on which you begin the Eraser stroke. If you begin erasing from an empty point, nothing is erased. Strokes are unaffected by the eraser in this mode.

- 3 Click the Eraser Shape modifier and select an eraser shape and size. Make sure that the Faucet modifier is not selected.
- 4 Drag on the Stage.

Transforming objects

You can transform graphic objects, as well as groups, text blocks, and instances, by using the Free Transform tool or the options in the **Modify > Transform** menu. Depending on the type of element you select, you can transform, rotate, skew, scale, or distort the element. You can change or add to a selection during a transformation operation.

When you transform an object, group, text box, or instance, the Property inspector for that item displays any changes made to the item's dimensions or position.

A bounding box appears during transform operations that involve dragging. The bounding box is rectangular (unless it was modified with the Distort command or the Envelope modifier), with its edges initially aligned parallel to the edges of the Stage. Transformation handles are located on each corner and in the middle of each side. As you drag, the bounding box previews the transformations.

Move, realign, change, and track the transformation point

During a transformation, a transformation point appears at the center of a selected element. The transformation point is initially aligned with the object's center point. You can move the transformation point, return it to its default location, and move the default point of origin.

For scaling, skewing, or rotating graphic objects, groups, and text blocks, the point opposite the point you drag is the point of origin by default. For instances, the transformation point is the point of origin by default. You can move the default point of origin for a transformation.

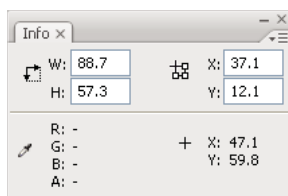
- 1 Select the Free Transform tool , or select one of the Modify > Transform commands.

Once you have begun a transformation, you can track the location of the transformation point in the Info panel and in the Property inspector.

- 2 Do one of the following:

- To move the transformation point, drag it from within the selected graphic object.
- To realign the transformation point with the element's center point, double-click the transformation point.
- To switch the point of origin for a scale or skew transformation, hold down the Alt key (Windows) or Option key (Macintosh) while dragging your chosen object control point during the transformation.
- To display the transformation point coordinates in the Info panel, click the Registration/Transformation Point button in the Info panel. The lower-right square in the button becomes a circle to indicate the registration point coordinates are being displayed.

When you select the center square, the X and Y values to the right of the coordinate grid in the Info panel display the x and y coordinates of the transformation point. In addition, the X and Y values for the transformation point appear in the Property inspector for the symbol.



Coordinate grid; Info panel with Registration/Transformation Point button in transformation mode, and with x and y coordinates of the selection transformation point visible.


By default, the Registration/Transformation Point button is in registration mode, and the X and Y values display the location of the upper-left corner of the current selection, relative to the upper-left corner of the Stage.

Note: For symbol instances, the X and Y values display the location of the symbol registration point, or the location of the upper-left corner of the symbol instance.

Use the Free Transform tool

You can perform individual transformations or combine several transformations, such as moving, rotating, scaling, skewing, and distortion.

Note: The Free Transform tool cannot transform symbols, bitmaps, video objects, sounds, gradients, or text. If a multiple selection contains any of these items, only the shape objects are distorted. To transform a text block, first convert the characters to shape objects.

- 1 Select a graphic object, group, instance, or text block on the Stage.
- 2 Click the Free Transform tool .

Moving the pointer over and around the selection changes the pointer to indicate which transformation function is available.

- 3 To transform the selection, drag the handles:
 - To move the selection, position the pointer over the object within the bounding box, and drag the object to a new position. Do not drag the transformation point.
 - To set the center of rotation or scaling, drag the transformation point to a new location.
 - To rotate the selection, position the pointer just outside a corner handle and drag. The selection rotates around the transformation point. Shift-drag to rotate in 45° increments.
 - To rotate around the opposite corner, Alt-drag (Windows) or Option-drag (Macintosh).
 - To scale the selection, drag a corner handle diagonally to scale in two dimensions. Shift-drag to resize proportionally.
 - To scale in the respective direction only, drag a corner handle or a side handle horizontally or vertically.
 - To skew the selection, position the pointer on the outline between the transformation handles and drag.
 - To distort shapes, press Control (Windows) or Command (Macintosh) and drag a corner handle or a side handle.
 - To *taper* the object—to move the selected corner and the adjoining corner equal distances from their origins, Shift-Control-click to drag (Windows) or Shift-Command-click to drag (Macintosh) a corner handle.
- 4 To end the transformation, click outside the selected item.

Distort objects

When you apply a Distort transformation to a selected object, dragging a corner handle or an edge handle on the bounding box moves the corner or edge and realigns the adjoining edges. Shift-drag a corner point to constrain the distortion to a taper—that is, move that corner and the adjoining corner an equal distance and in the opposite direction from each other. The adjoining corner is the corner on the same axis as the direction you drag. Control-click (Windows) or Command-click (Macintosh) to drag a middle point on an edge to move the entire edge freely.

You can distort graphic objects by using the Distort command. You can also distort objects when performing a free transform on them.

Note: The Distort command cannot modify symbols, shape primitives, bitmaps, video objects, sounds, gradients, object groups, or text. If a multiple selection contains any of these items, only the shape objects are distorted. To modify text, first convert the characters to shape objects.

- 1 Select a graphic object or objects on the Stage.
- 2 Select Modify > Transform > Distort.
- 3 Place the pointer on one of the transformation handles and drag.
- 4 To end the transformation, click outside the selected object or objects.

Modify shapes with the Envelope modifier

The Envelope modifier lets you warp and distort objects. An envelope is a bounding box that contains one or more objects. Changes made to an envelope's shape affect the shape of the objects in the envelope. You edit the shape of an envelope by adjusting its points and tangent handles.

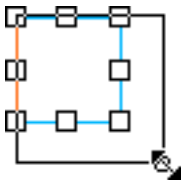
Note: The Envelope modifier cannot modify symbols, bitmaps, video objects, sounds, gradients, object groups, or text. If a multiple selection contains any of these items, only the shape objects are distorted. To modify text, first convert the characters to shape objects.

- 1 Select a shape on the Stage.
- 2 Select Modify > Transform > Envelope.
- 3 Drag the points and tangent handles to modify the envelope.

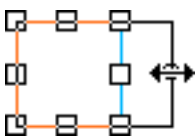
Scale objects

Scaling an object enlarges or reduces the object horizontally, vertically, or both.

- 1 Select a graphic object or objects on the Stage.
- 2 Select Modify > Transform > Scale.
- 3 Do one of the following:
 - To scale the object both horizontally and vertically, drag one of the corner handles. Proportions are maintained as you scale. Shift-drag to scale nonuniformly.



- To scale the object either horizontally or vertically, drag a center handle.



- 4 To end the transformation, click outside the selected object or objects.

Note: When you increase the size of a number of items, items near the edges of the bounding box might be moved off the Stage. If this occurs, select View > Pasteboard to see the elements that are beyond the edges of the Stage.

See also


[“About 9-slice scaling and movie clip symbols”](#) on page 166

[“Edit movie clip symbols with 9-slice scaling”](#) on page 167

Rotate and skew objects

Rotating an object turns it around its transformation point. The transformation point is aligned with the registration point, which defaults to the center of the object, but you can move the point by dragging it.

You can rotate an object by the following methods:

- Dragging with the Free Transform tool  (you can skew and scale the object in the same operation).
- By specifying an angle in the Transform panel (you can scale the object in the same operation).

Rotate and skew objects by dragging

- 1 Select the object or objects on the Stage.
- 2 Select Modify > Transform > Rotate And Skew.
- 3 Do one of the following:
 - Drag a corner handle to rotate the object.
 - Drag a center handle to skew the object.
- 4 To end the transformation, click outside the selected object or objects.

Rotate objects by 90°

- 1 Select the object or objects.
- 2 Select Modify > Transform > Rotate 90° CW to rotate clockwise, or Rotate 90° CCW to rotate counterclockwise.

Skew objects

Skewing an object transforms it by slanting it along one or both axes. You can skew an object by dragging or by entering a value in the Transform panel.

- 1 Select the object or objects.
- 2 Select Window > Transform.
- 3 Click Skew.
- 4 Enter angles for the horizontal and vertical values.

Flip objects

You can flip objects across their vertical or horizontal axis without moving their relative position on the Stage.

- 1 Select the object.
- 2 Select Modify > Transform > Flip Vertical or Flip Horizontal.

Restore transformed objects

When you use the Transform panel to scale, rotate, and skew instances, groups, and fonts, Flash saves the original size and rotation values with the object. This process lets you remove the transformations you applied and restore the original values.

You can undo only the most recent transformation performed in the Transform panel when you select Edit > Undo. You can reset all transformations performed in the Transform panel by clicking the Remove Transform button in the panel before you deselect the object.

Restore a transformed object to its original state

- 1 Select the transformed object.
- 2 Select Modify > Transform > Remove Transform.

Reset a transformation performed in the Transform panel

❖ With the transformed object still selected, click the Remove Transform button  in the Transform panel.

Combining objects

To create new shapes by combining or altering existing objects, use the Combine Objects commands in the Modify menu (Modify > Combine Objects). In some cases, the stacking order of selected objects determines how the operation works.

Each command applies to specific types of graphic objects, which are noted below. A merge shape is a shape drawn with a tool set to Merge Drawing mode. A drawing Object is a shape drawn with a tool set to Object Drawing mode.

The Combine Objects commands are:

Union Joins two or more merge shapes or drawing objects. The result is a single Object Drawing mode shape consisting of all the portions visible on the shapes before they were unified. The unseen, overlapping portions of the shapes are deleted.

Note: Unlike when you use the Group command (Modify > Group), you cannot break apart shapes joined by using the Union command.

Intersect Creates an object from the intersection of two or more drawing objects. The resulting Object Drawing shape consists of the overlapping portions of the combined shapes. Any part of the shape that doesn't overlap is deleted. The resulting shape uses the fill and stroke of the top-most shape in the stack.

Punch Removes portions of a selected drawing object as defined by the overlapping portions of another selected drawing object positioned in front of it. Any part of a drawing object that is overlapped by the top-most object is deleted, and the top-most object is deleted entirely. The resulting objects remain separate and are not combined into a single object (unlike the Union or Intersect commands, which join the objects together).

Crop Uses the outline of one drawing object to crop another drawing object. The front or top-most object defines the shape of the cropped area. Any part of an underlying drawing object that overlaps with the top-most object remains, while all other portions of the underlying objects are deleted, and the top-most object is deleted entirely. The resulting objects remain separate, and are not combined into a single object (unlike the Union or Intersect commands, which join the objects).

Moving, arranging, and deleting artwork

Organizing and laying out your artwork in Flash is made easy with tools that enable you to select, position, and stack objects precisely. Tools are available that let you measure and align objects; group objects so that they are treated as a single unit; and selectively isolate, lock, or hide objects.

Selecting objects

To modify an object, select it first. Select objects with the Pointer, Subselection, and Lasso tools. You can group individual objects to manipulate them as a single object. Modifying lines and shapes can alter other lines and shapes on the same layer. When you select objects or strokes, Flash highlights them with a marquee.

You can choose to select only an object's strokes or only its fills. You can hide selection highlighting to edit objects without viewing highlighting.

When you select an object, the Property inspector displays the following:

- The object's stroke and fill, its pixel dimensions, and the x and y coordinates of the object's transformation point

- A mixed selection, if you select multiple items. The pixel dimensions and x and y coordinates of the selected set of items.

You can use a shape's Property inspector to change that object's stroke and fill.

To prevent a group or symbol from being selected and accidentally changed, lock the group or symbol.

See also


[“Creating and Editing Artwork”](#) on page 91

[“Colors, gradients, and strokes”](#) on page 126

[“Group objects”](#) on page 121

[“About symbols”](#) on page 147

Select objects with the Selection tool

The Selection tool  lets you select entire objects by clicking an object or dragging to enclose the object within a rectangular selection marquee.

Note: To select the Selection tool, you can also press the V key. To temporarily switch to the Selection tool when another tool is active, hold down the Control key (Windows) or Command key (Macintosh).


To disable the Shift-selecting option, deselect the option in Flash General Preferences. See [“Set preferences in Flash”](#) on page 37. Instances, groups, and type blocks must be completely enclosed to be selected.

- To select a stroke, fill, group, instance, or text block, click the object.
- To select connected lines, double-click one of the lines.
- To select a filled shape and its stroked outline, double-click the fill.
- To select objects within a rectangular area, drag a marquee around the object or objects to select.
- To add to a selection, hold down the Shift key while making additional selections.
- To select everything on every layer of a scene, select Edit > Select All, or press Control+A (Windows) or Command+A (Macintosh). Select All doesn't select objects on locked or hidden layers, or layers not on the current Timeline.
- To deselect everything on every layer, select Edit > Deselect All, or press Control+Shift+A (Windows) or Command+Shift+A (Macintosh).
- To select everything on one layer between keyframes, click a frame in the Timeline.
- To lock or unlock a group or symbol, select the group or symbol, and then select Modify > Arrange > Lock. Select Modify > Arrange > Unlock All to unlock all locked groups and symbols.

Draw a freehand selection area

- 1 Drag the Lasso tool  around the area.
- 2 End the loop approximately where you started, or let Flash automatically close the loop with a straight line.

Draw a straight-edged selection area

- 1 Select the Lasso tool's Polygon Mode modifier  in the options area of the Tools panel.
- 2 Click to set the starting point.
- 3 Position the pointer where you want the first line to end, and click. Continue setting end points for additional line segments.

- 4 To close the selection area, double-click.

Draw a selection area with both freehand and straight-line edges

When you use the Lasso tool and its Polygon Mode modifier, you can switch between the freehand and straight-edged selection modes.

- 1 Deselect the Polygon Mode modifier of the Lasso tool.
- 2 To draw a freehand segment, drag the Lasso tool on the Stage.
- 3 To draw straight-edged segments, hold down the Alt key (Windows) or Option key (Macintosh) and click to set start and end points for each new line segment.
- 4 To close the selection area, do one of the following:
 - Release the mouse button; Flash will close the selection area for you.
 - Double-click on the starting end of the selection area line.

Hide selection highlighting

Hiding highlighting while you are selecting and editing objects lets you see how artwork appears in its final state.

- ❖ Select View > Hide Edges.

Select the command again to show selection highlighting.

Set custom bounding box colors for selected objects

You can set different colors to be used for the bounding box rectangles that appear around different kinds of selected objects on the Stage.

- 1 Select Edit > Preferences (Windows) or Flash > Preferences (Macintosh).
- 2 Click the General category.
- 3 In the Highlight Color section, select a color for each type of object and click OK.

Set preferences for selecting

The Pointer, Subselection, and Lasso tools select objects by clicking on them. The Pointer and Subselection tools select objects by dragging a rectangular selection marquee around the object. The Lasso tool selects objects by dragging a free-form selection marquee around the object. When an object is selected, a rectangular box appears around the object.

- 1 Select Edit > Preferences (Windows) or Flash > Preferences (Macintosh).
- 2 In the General category of the Preferences dialog box, do one of the following:
 - To select only objects and points that are completely enclosed by the selection marquee, deselect Contact-Sensitive Selection and Lasso tools. Points that lie within the selection area will still be selected.
 - To select objects or groups that are only partially enclosed by the selection marquee, select Contact-Sensitive Selection and Lasso tools.

Arranging objects

Stack objects

In a layer, Flash stacks objects in the order in which they are created, placing the most recently created object at the top of the stack. The stacking order of objects determines how they appear when they overlap. You can change the stacking order of objects at any time.

Drawn lines and shapes always appear below groups and symbols on the stack. To move them up the stack, you must group them or make them into symbols.

Layers also affect the stacking order. Everything on Layer 2 appears in front of everything on Layer 1, and so on. To change the order of layers, drag the layer name in the Timeline to a new position.

- 1 Select the object.
- 2 Do one of the following:
 - Select **Modify > Arrange > Bring To Front Or Send To Back** to move the object or group to the top or bottom of the stacking order.
 - Select **Modify > Arrange > Bring Forward Or Send Backward** to move the object or group forward or backward one position in the stacking order.

If more than one group is selected, the groups move in front of or behind all unselected groups, while maintaining their order relative to each other.

See also

“[Create and organize layers](#)” on page 173

Align objects

The Align panel lets you align selected objects along the horizontal or vertical axis. You can align objects vertically along the right edge, center, or left edge of the selected objects, or horizontally along the top edge, center, or bottom edge of the selected objects.

For a tutorial about the layout tools in Flash, see [Use Layout Tools on the Flash Tutorials page at www.adobe.com/go/learn_fl_tutorials](#).

- 1 Select the objects to align.
- 2 Select **Window > Align**.
- 3 To apply alignment modifications relative to Stage dimensions, in the Align panel, select **To Stage**.
- 4 To modify the selected object(s), select the alignment buttons.

Group objects

To manipulate elements as a single object, group them. For example, after creating a drawing, you might group the elements of the drawing so that you can easily select and move the drawing as a whole.

When you select a group, the Property inspector displays the *x* and *y* coordinates of the group and its pixel dimensions.

You can edit groups without ungrouping them. You can also select an individual object in a group for editing without ungrouping the objects.

- ❖ Select the objects to group. You can select shapes, other groups, symbols, text, and so on.
 - To group objects, select **Modify > Group**, or press **Control+G** (Windows) or **Command+G** (Macintosh).

- To ungroup objects, select Modify > Ungroup, or press Control+Shift+G (Windows) or Command+Shift+G (Macintosh).

Edit a group or an object within a group

1 Select the group, and then select Edit > Edit Selected, or double-click the group with the Selection tool.

Everything on the page that is not part of the group is dimmed, indicating that elements outside the group are inaccessible.

2 Edit any element within the group.

3 Select Edit > Edit All, or double-click a blank spot on the Stage with the Selection tool.

Flash restores the group to its status as a single entity, and you can work with other elements on the Stage.

Break apart groups and objects

To separate groups, instances, and bitmaps into ungrouped, editable elements, you break them apart, which significantly reduces the file size of imported graphics.

Although you can select Edit > Undo immediately after breaking apart a group or object, breaking apart is not entirely reversible. It affects objects as follows:

- Severs a symbol instance's link to its master symbol
- Discards all but the current frame in an animated symbol
- Converts a bitmap to a fill
- Places each character into a separate text block when applied to text blocks
- Converts characters to outlines when applied to a single text character.

Do not confuse the Break Apart command with the Ungroup command. The Ungroup command separates grouped objects, returning grouped elements to the state they were in before grouping. It does not break apart bitmaps, instances, or type, or convert type to outlines.

1 Select the group, bitmap, or symbol to break apart.

2 Select Modify > Break Apart.

***Note:** Breaking apart animated symbols, or groups in an interpolated animation is not recommended and might have unpredictable results. Breaking apart complex symbols and large blocks of text can take a long time. You might need to increase the application's memory allocation to properly break apart complex objects.*

See also

["Break text apart"](#) on page 252

Snapping art into position

To automatically align graphic elements with one another, use *snapping*. Flash provides three ways for you to align objects on the Stage:

- Object snapping snaps objects directly to other objects along their edges.
- Pixel snapping snaps objects directly to individual pixels or lines of pixels on the Stage.
- Snap alignment snaps objects to a specified *snap tolerance*, a preset boundary between objects and other objects or between objects and the edge of the Stage.

***Note:** You can also snap to the grid or to guides.*

See also

[“About the main toolbar and edit bar”](#) on page 20

[“Drawing preferences”](#) on page 96

Turn object snapping on or off

To turn on object snapping, use the Snap To Objects modifier for the Selection tool, or the Snap To Objects command in the View menu.

If the Snap To Objects modifier for the Selection tool is on, a small black ring appears under the pointer when you drag an element. The small ring changes to a larger ring when the object is within snapping distance of another object.

❖ Select View > Snapping > Snap To Objects. A check mark appears next to the command when it is on.

When you move or reshape an object, the position of the Selection tool on the object provides the reference point for the snap ring. For example, if you move a filled shape by dragging near its center, the center point snaps to other objects. This is particularly useful for snapping shapes to motion paths for animating.

Note: For better control of object placement when snapping, begin dragging from a corner or center point.

Adjust object snapping tolerances

- 1 Select Edit > Preferences (Windows) or Flash > Preferences (Macintosh), and click Drawing.
- 2 Under Drawing Settings, adjust the Connect Lines setting.

Use pixel snapping

To turn on pixel snapping, use the Snap To Pixels command in the View menu. If Snap To Pixels is on, a pixel grid appears when the view magnification is set to 400% or higher. The pixel grid represents the individual pixels that appear in your Flash application. When you create or move an object, it is constrained to the pixel grid.

If you create a shape whose edges fall between pixel boundaries—for example, if you use a stroke with a fractional width, such as 3.5 pixels—Snap To Pixels snaps to pixel boundaries, not to the edge of the shape.

- To turn pixel snapping on or off, select View > Snapping > Snap To Pixels. If the magnification is set to 400% or higher, a pixel grid is displayed. A check mark appears next to the command when it is on.
- To turn pixel snapping on or off temporarily, press the C key. When you release the C key, pixel snapping returns to the state you selected with View > Snapping > Snap To Pixels.
- To temporarily hide the pixel grid, press the X key. When you release the X key, the pixel grid reappears.

Select settings for Snap Alignment

When you select Snap Alignment settings, set the snap tolerance between horizontal or vertical edges of objects, and between objects' edges and the Stage border. You can also turn on snap alignment between the horizontal and the vertical centers of objects. All Snap Alignment settings are measured in pixels.

- 1 Select View > Snapping > Edit Snapping.
- 2 In the Edit Snapping dialog box, select the types of objects to want to snap to.
- 3 Click the Advanced button and select any of the following:
 - To set the snap tolerance between objects and the Stage border, enter a value for Movie Border.
 - To set the snap tolerance between the horizontal or vertical edges of objects, enter a value for Horizontal, Vertical, or both.
 - To turn on Horizontal or Vertical Center Alignment, select Horizontal or Vertical Center Alignment or both.


Turn on Snap Alignment

When Snap Alignment is turned on, dotted lines appear on the Stage when you drag an object to the specified snap tolerance. For example, if you set Horizontal snap tolerance to 18 pixels (the default setting), a dotted line appears along the edge of the object you are dragging when the object is exactly 18 pixels from another object. If you turn on Horizontal Center Alignment, a dotted line appears along the horizontal center vertices of two objects when you precisely align the vertices.

- ❖ Select View > Snapping > Snap Align. A check mark appears next to the command when it is on.

Moving and copying objects

Move objects by dragging

- 1 Select an object or multiple objects.
- 2 Select the Selection tool , position the pointer over the object, and do one of the following:
 - To move the object, drag it to the new position.
 - To copy the object and move the copy, Alt-drag (Windows) or Option-drag (Macintosh).
 - To constrain the object's movement to multiples of 45°, Shift-drag.

Move objects by using the arrow keys

- 1 Select an object or multiple objects.
- 2 Do one of the following:
 - To move the selection 1 pixel at a time, press the arrow key for the direction in which you want the object to move.
 - To move the selection 10 pixels at a time, press Shift+arrow key.

Note: When Snap To Pixels is selected, the arrow keys move objects by pixel increments on the document's pixel grid, not by pixels on the screen.

Move objects by using the Property inspector

- 1 Select an object or multiple objects.
- 2 If the Property inspector is not visible, select Window > Properties.
- 3 Enter x and y values for the location of the upper-left corner of the selection.

The units are relative to the upper-left corner of the Stage.

Note: The Property inspector uses the units specified for the Ruler Units option in the Document Properties dialog box.

Move objects by using the Info panel

- 1 Select an object or multiple objects.
- 2 If the Info Panel is not visible, select Window > Info.
- 3 Enter x and y values for the location of the upper-left corner of the selection.

The units are relative to the upper-left corner of the Stage.

Move and copy objects by pasting

To move or copy objects between layers, scenes, or other Flash files, use the pasting technique. You can paste an object in a position relative to its original position.

- 1 Select an object or multiple objects.
- 2 Select Edit > Cut or Edit > Copy.
- 3 Select another layer, scene, or file, and then select Edit > Paste In Place to paste the selection in the same position relative to the Stage. Select Edit > Paste In Center to paste the selection in the center of the work area.

Copying objects with the clipboard

Elements copied to the clipboard are anti-aliased, so they look as good in other applications as they do in Flash. This feature is useful for frames that include a bitmap image, gradients, transparency, or a mask layer.


Graphics pasted from other Flash documents or programs are placed in the current frame of the current layer. How a graphic element is pasted into a Flash scene depends on the type of element it is, its source, and the preferences you have set:

- Text from a text editor becomes a single text object.
- Vector-based graphics from any drawing program become a group that can be ungrouped and edited.
- Bitmaps become a single grouped object just like imported bitmaps. You can break apart pasted bitmaps or convert pasted bitmaps to vector graphics.

Note: Before pasting graphics from Illustrator into Flash, convert colors to RGB in Illustrator.

Copy transformed objects

You can create a scaled, rotated, or skewed copy of an object.

- 1 Select an object.
- 2 Select Window > Transform.
- 3 Enter scale, rotation, or skew values.
- 4 Click the Duplicate Selection and Transform button  in the Transform panel.

Delete objects

Deleting an object removes it from the file. Deleting an instance of an object on the Stage does not delete the symbol from the library.

- 1 Select an object or multiple objects.
- 2 Do one of the following:
 - Press Delete or Backspace.
 - Select Edit > Clear.
 - Select Edit > Cut.
 - Right-click (Windows) or Control-click (Macintosh) the object, and select Cut from the context menu.

See also

[“Scale objects”](#) on page 116

[“Rotate and skew objects”](#) on page 116

“Create a new document” on page 43

“Convert a bitmap to a vector graphic” on page 89

“Use pixel snapping” on page 123

Colors, gradients, and strokes

Color models describe the colors we see and work with in digital graphics. Each color model, such as RGB, HSB, or CMYK, represents a different method for describing and classifying color. Color models use numeric values to represent the visible spectrum of color. A color space is a variant of a color model and has a specific gamut (range) of colors. For example, within the RGB color model are a number of color spaces: Adobe® RGB, sRGB, and Apple® RGB. While each of these color spaces defines color using the same three axes (R, G, and B), their gamuts are different.

When you work with the colors in a graphic, you are actually adjusting numerical values in the file. It’s easy to think of a number as a color, but these numerical values are not absolute colors in themselves—they only have a color meaning within the color space of the device that is producing the color.

Because each device has its own color space, it can reproduce colors only in its gamut. When an image moves from one device to another, image colors may change because each device interprets the RGB or HSB values according to its own color space. For example, it is impossible for all the colors viewed on a monitor to be identically matched in a print from a desktop printer. A printer operates in a CMYK color space and a monitor operates in an RGB color space. Their gamuts are different. Some colors produced by inks cannot be displayed on a monitor, and some colors that can be displayed on a monitor cannot be reproduced using inks on paper.

When creating colors for use in Flash documents, keep in mind that even though it is impossible to perfectly match all colors on different devices, you can achieve good results by considering the graphic display capabilities of the devices in use by your target audience.

Adobe® Flash® CS4 Professional lets you apply, create, and modify colors using the RGB or HSB color models. Using the default palette or a palette you create, you can choose colors to apply to the stroke or fill of an object you are about to create, or an object already on the Stage.

When applying a stroke color to a shape, you can do any of the following:

- Apply a solid color, gradient, or bitmap to a shape’s fill. To apply a bitmap fill to a shape, you must import a bitmap into the current file. Select any solid color, gradient, and the style and weight of the stroke.
- Create an outlined shape with no fill by using No Color as a fill.
- Create a filled shape with no outline by using No Color as an outline.
- Apply a solid color fill to text.

With the Color panel, you can create and edit solid colors and gradient fills in RGB and HSB modes.

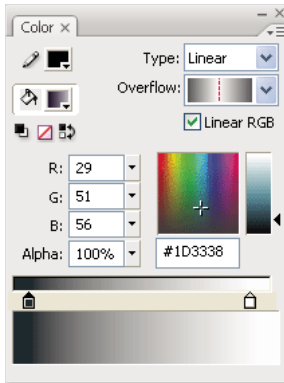
To access the system color picker, Alt-double-click (Windows) or Option-double-click (Macintosh) the Stroke Color or Fill Color control in the Color panel, or select the Color Picker icon  from the Stroke Color or Fill Color control in the Tools panel or Shape Property inspector.

The Color panel

The Color panel lets you modify the color palette of a FLA and change the color of strokes and fills, including the following:

- Import, export, delete, and otherwise modify the color palette for a FLA file by using the Swatches panel.

- Select colors in hexadecimal mode.
- Create multicolor gradients.
- Use gradients to produce a wide range of effects, such as giving an illusion of depth to a two-dimensional object.



The Color panel with the gradient controls displayed.

The Color panel contains the following controls:

Stroke Color Changes the color of the stroke, or the border, of a graphic object.

Fill Color Changes the color of the fill. The fill is the area of color that fills up the shape.

Type Menu Changes the fill style:

- **None** Removes the fill.
- **Solid** Provides a solid, single fill color.
- **Linear** Produces a gradient that blends on a linear path.
- **Radial** Produces a gradient that blends outward in a circular path from a central focal point.
- **Bitmap** Tiles the selected fill area with a bitmap image that you can select. When you choose Bitmap, a dialog box lets you select a bitmap image on your local computer, and add it to the library. You can apply this bitmap as a fill; the appearance is similar to a mosaic tile pattern with the image repeated within the shape.

RGB Lets you change the density of the red, green, and blue (RGB) colors in a fill.

Alpha Sets the opacity for a solid fill, or the currently selected slider for a gradient fill. An alpha value of 0% creates an invisible (or transparent) fill; an alpha value of 100% creates an opaque fill.

Current Color Swatch Displays the currently selected color. If you select a gradient fill type (Linear or Radial) from the fill Type menu, the Current Color Swatch displays the color transitions within the gradient you create.

System Color Picker Lets you select a color visually. Click System Color Picker and drag the cross-hair pointer until you find the color you want.

Hexadecimal value Displays the current color's hexadecimal value. To change the color using the hexadecimal value, type in a new value. Hexadecimal color values (also called hex values) are 6-digit alphanumeric combinations that represent a color.

Overflow Lets you control colors applied past the limits of a linear or radial gradient.

- **Extend** (Default) Applies the colors you specify past the end of the gradient.
- **Reflect** Causes the gradient colors to fill the shape using a reflective mirroring effect. The gradients you specify are repeated in a pattern from the beginning of the gradient to the end, and then repeated in the opposite sequence from

the end of the gradient to the beginning, and then back to the beginning of the gradient to the end until the selected shape is filled.

- **Repeat** Repeats the gradient from the beginning of the gradient to the end until the selected shape is filled.

Note: Overflow modes are supported only in Adobe Flash Player 8 and later.

Linear RGB Creates a SVG-compliant (Scalable Vector Graphics) linear or radial gradient.

Color palettes

Each Flash file contains its own color palette, stored in the Flash document. Flash displays a file's palette as swatches in the Fill Color and Stroke Color controls and in the Swatches panel. The default color palette is the web-safe palette of 216 colors. To add colors to the current color palette, use the Color panel.

You can import and export both solid and gradient color palettes between Flash files, as well as between Flash and other applications.

See also

[“Create or edit a solid color”](#) on page 129

The default palette and the web-safe palette

Save the current palette as the default palette, replace the current palette with the default palette defined for the file, or load the web-safe palette to replace the current palette.

- To load or save the default palette, in the Swatches panel, select one of the following commands from the menu in the upper-right corner:

Load Default Colors Replaces the current palette with the default palette.

Save As Default Saves the current color palette as the default palette. The new default palette is used when you create new files.

- To load the web-safe 216-color palette, in the Swatches panel, select Web 216 from the menu in the upper-right corner

Sort color by hue in the palette

To make it easier to locate a color, sort colors in the palette by hue.

- ❖ In the Swatches panel, select Sort by Color from the menu in the upper-right corner.

Import and export color palettes

To import and export both RGB colors and gradients between Flash files, use Flash Color Set files (CLR files). Import and export RGB color palettes by using Color Table files (ACT files). You can also import color palettes, but not gradients, from GIF files. You cannot import or export gradients from ACT files.

Import a color palette

1 In the Swatches panel, select one of the following commands from the menu in the upper-right corner:

- To append the imported colors to the current palette, select Add Colors.
- To replace the current palette with the imported colors, select Replace Colors.

2 Navigate to the desired file, select it, and click OK.

Export a color palette

- 1 In the Swatches panel, select Save Colors from the menu in the upper-right corner and enter a name for the color palette.
- 2 For Save As Type (Windows) or Format (Macintosh), select Flash Color Set or Color Table. Click Save.

Create or edit a solid color




You can create any color using the Color panel. If an object is selected on the Stage, the color modifications you make in the Color panel are applied to the selection. You can select colors in RGB or HSB, or you can expand the panel to use hexadecimal mode. You can also specify an alpha value to define the degree of transparency for a color. In addition, you can select a color from the existing color palette.

You can expand the Color panel to display a larger color space in place of the color bar, a split color swatch showing the current and previous colors, and a Brightness slider to modify color brightness in all color modes.

- 1 To apply the color to existing artwork, select an object or objects on the Stage, and select Window > Color.
- 2 To select a color mode display, select RGB (the default setting) or HSB from the panel menu in the upper-right corner.
- 3 Click the Stroke or Fill icon to specify which attribute to modify.
Note: Click the icon, not the Color control, or the Color Picker opens.
- 4 If you selected the Fill icon in step 3, verify that Solid is selected in the Type menu.
- 5 If an object is selected on the Stage, the color modifications you make in the Color panel are applied to the selection. Do one of the following:

- To select a color, click in the color space in the Color panel. To adjust the brightness of the color, drag the Brightness slider.

Note: To create colors other than black or white, make sure the Brightness slider is not set to either extreme.

- Enter values in the color value boxes: Red, Green, and Blue values for RGB display; Hue, Saturation, and Brightness values for HSB display; or hexadecimal values for hexadecimal display. Enter an Alpha value to specify the degree of transparency, from 0 for complete transparency to 100 for complete opacity.
- To return to the default color settings, black and white (black stroke and white fill), click the Black and White button .
- To swap colors between the fill and the stroke, click the Swap Colors button .
- To apply no color to the fill or stroke, click the No Color button .

Note: You cannot apply a stroke or fill of No Color to an existing object. Instead, select the existing stroke or fill, and delete it.

- Click the Stroke or Fill Color control, and select a color.
- 6 To add the new color to the color swatch list for the current document, select Add Swatch from the menu in the upper-right corner.

Create or edit a gradient fill

A gradient is a multicolor fill in which one color gradually changes into another color. Flash lets you apply up to 15 color transitions to a gradient. Creating a gradient is a good way to create a smooth color gradation across one or more objects. You can save a gradient as a swatch to make it easy to apply the gradient to multiple objects. Flash can create two types of gradients:

Linear gradients change color along a single axis (horizontal or vertical).

Radial gradients change color in an outward direction starting from a central focal point. You can adjust the direction of a gradient, its colors, the location of the focal point, and many other properties of the gradient.

Adobe® Flash® CS4 Professional provides additional control over linear and radial gradients for use with Flash Player. These controls, called overflow modes, let you specify how colors are applied beyond the gradient.

For a sample of gradients, see the Flash Samples page at www.adobe.com/go/learn_fl_samples. Download and decompress the Samples zip file and navigate to the Graphics\AnimationAndGradients folder to access the sample.

- 1 To apply a gradient fill to existing artwork, select an object or objects on the Stage.
- 2 If the Color panel is not visible, select Window > Color.
- 3 To select a color display mode, select RGB (the default setting) or HSB from the panel menu.
- 4 Select a gradient type from the Type menu:
 - Linear** Creates a gradient that shades from the starting point to the end point in a straight line.
 - Radial** Produces a gradient that blends outward in a circular path from a central focal point.

Note: When you select a linear or radial gradient, the Color panel also includes two other options if you are publishing for Flash Player 8 or later. First, the Overflow menu is enabled below the Type menu. Use the Overflow menu to control the colors applied past the limits of the gradient. Second, the gradient definition bar appears, with pointers below the bar indicating the colors in the gradient.
- 5 (Optional) From the Overflow menu, select an overflow mode to apply to the gradient: Extend (the default mode), Reflect, or Repeat.
- 6 (Optional) To create an SVG-compliant (Scalable Vector Graphics) linear or radial gradient, select the Linear RGB check box. This will allow the gradient to look smooth when scaled to different sizes after it is first applied.
- 7 To change a color in the gradient, select one of the color pointers below the gradient definition bar (the triangle at the top of the selected color pointer will turn black). Then click in the color space pane that appears above the gradient bar. Drag the Brightness slider to adjust the lightness of the color.
- 8 To add a pointer to the gradient, click on or below the gradient definition bar. Select a color for the new pointer, as described in the previous step.

You can add up to 15 color pointers, letting you create a gradient with up to 15 color transitions.
- 9 To reposition a pointer on the gradient, drag the pointer along the gradient definition bar. Drag a pointer down and off of the gradient definition bar to remove it.
- 10 To save the gradient, click the triangle in the upper-right corner of the Color panel, and select Add Swatch from the menu.

The gradient is added to the Swatches panel for the current document.
- 11 To transform the gradient, such as to make a vertical gradient instead of a horizontal one, use the Gradient Transform tool. See “[Transform gradient and bitmap fills](#)” on page 134 for more information.

Adjust Stroke and Fill color

You can specify the stroke and fill color of graphic objects and shapes using either the Stroke Color and Fill Color controls in the Tools panel, or the Stroke Color and Fill Color controls in the Property inspector.

The Stroke Color and Fill Color section of the Tools panel contains controls for activating the Stroke Color and Fill Color boxes, which in turn determine whether the strokes or fills of selected objects are affected by color choices. Also, the Colors section has controls for quickly resetting colors to the default, setting the stroke and fill color settings to None, and swapping fill and stroke colors.

In addition to letting you select a stroke and fill color for a graphic object or shape, the Property inspector provides controls for specifying the stroke width and style.

To use these controls to change the painting attributes of existing objects, first select the objects on the Stage.

Adjust stroke and fill color using the Tools panel

The Tools panel Stroke Color and Fill Color controls set the painting attributes of new objects you create with the drawing and painting tools. To use these controls to change the painting attributes of existing objects, first select the objects on the Stage.

- Click the Stroke or Fill Color control, and select a color swatch.
- Click the System Color Picker button in the pop-up window, and select a color.
- Type a color's hexadecimal value in the box.
- To return to the default color settings (white fill and black stroke), click the Black And White button in the Tools panel.
- To remove any stroke or fill, click the No Color button.

Note: The No Color button appears only when you are creating an oval or rectangle. You can create an object without a stroke or fill, but you cannot use the No Color button with an existing object. Instead, select the existing stroke or fill and delete it.

- To Swap colors between the fill and the stroke, click the Swap Colors button in the Tools panel.

Apply a solid color fill using the Property inspector

- 1 Select a closed object or objects on the Stage.
- 2 Select Window > Properties.
- 3 To select a color, click the Fill Color control and do one of the following:
 - Select a color swatch from the palette.
 - Type a color's hexadecimal value in the box.

Select a stroke color, style, and weight using the Property inspector

To change the stroke color, style, and weight for a selected object, use the Stroke Color control in the Property inspector. For stroke style, choose from styles that are preloaded with Flash, or create a custom style. To select a solid color fill, use the Fill Color control in the Property inspector.

- 1 Select an object or objects on the Stage (for symbols, first double-click to enter symbol-editing mode).
- 2 Select Window > Properties.
- 3 To select a stroke style, click the Style menu and select an option. To create a custom style, click Custom in the Property inspector, select options in the Stroke Style dialog box, and click OK.

Note: Selecting a stroke style other than Solid can increase file size.

- 4 To select a stroke weight, set the Stroke slider or enter a value in the text box.

5 To enable stroke hinting, select the Stroke Hinting check box. Stroke hinting adjusts line and curve anchors on full pixels, preventing blurry vertical or horizontal lines.

6 To set the style for a path end, select a Cap option:

None Is flush with the path's end.

Round Adds a round cap that extends beyond the path end by half the stroke width.

Square Adds a square cap that extends beyond the path by half the stroke width.

7 (Optional) If you are drawing lines using the Pencil or Brush tools with the drawing mode set to Smooth, use the Smoothing slider to specify the degree to which Flash smooths the lines you draw.

By default, the Smoothing value is set to 50, but you can specify a value from 0 to 100. The greater the smoothing value, the smoother the resulting line.

Note: When the drawing mode is set to Straighten or Ink, the Smoothing slider is disabled.

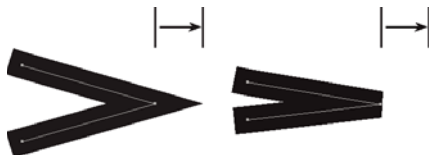
8 To define how two path segments meet, select a Join option. To change the corners in an open or closed path, select a path and select another join option.



Miter, round, and bevel joins.

9 To avoid beveling a Miter join, enter a Miter limit.

Line lengths exceeding this value are squared instead of pointed. For example, a Miter limit of 2 for a 3-point stroke means that when the length of the point is twice the stroke weight, Flash removes the limit point.



Applying a Miter limit.

Adjust the strokes of multiple lines or shapes

To change the stroke color, width, and style of one or more lines or shape outlines, use the Ink Bottle tool. You can apply only solid colors, not gradients or bitmaps, to lines or shape outlines.

Using the Ink Bottle tool, rather than selecting individual lines, makes it easier to change the stroke attributes of multiple objects at one time.

- 1 Select the Ink Bottle tool from the Tools panel.
- 2 Select a stroke color.
- 3 Select a stroke style and stroke width from the Property inspector.
- 4 To apply the stroke modifications, click an object on the Stage.

Copy strokes and fills

Use the Eyedropper tool to copy fill and stroke attributes from one object and immediately apply them to another object. The Eyedropper tool also lets you sample the image in a bitmap to use as a fill.

- 1 To apply the attributes of a stroke or filled area to another stroke or filled area, select the Eyedropper tool and click the stroke or filled area whose attributes you want to apply.

When you click a stroke, the tool automatically changes to the Ink Bottle tool. When you click a filled area, the tool automatically changes to the Paint Bucket tool with the Lock Fill modifier turned on.

- 2 Click another stroke or filled area to apply the new attributes.

See also

[“Break apart groups and objects”](#) on page 122

Duplicate, delete, and clear colors

Duplicate colors in the palette, delete individual colors, or clear all colors from the palette.

- To duplicate or delete a color, select Window > Swatches, click the color to duplicate or delete, and select Duplicate Swatch or Delete Swatch from the panel menu. When duplicating a swatch, the paint bucket appears. Click in the empty area of the Swatches panel with the paint bucket to make a duplicate of the selected color.
- To clear all colors from the color palette, in the Swatches panel, select Clear Colors from the panel menu. All colors except black and white are removed from the palette.

Modifying painted areas

The Paint Bucket tool fills enclosed areas with color. This tool lets you do the following:

- Fill empty areas, and change the color of already painted areas.
- Paint with solid colors, gradients, and bitmap fills.
- Use the Paint Bucket tool to fill areas that are not entirely enclosed.
- Have Flash close gaps in shape outlines as you use the Paint Bucket tool.

- 1 Select the Paint Bucket tool from the Tools panel.

- 2 Select a fill color and style.

- 3 Click the Gap Size modifier and select a gap size option:

- Don't Close Gaps to close gaps manually before filling the shape. Closing gaps manually can be faster for complex drawings.
- A Close option to have Flash fill a shape that has gaps.

Note: If gaps are too large, you might have to close them manually.

- 4 Click the shape or enclosed area to fill.


See also

[“Adjust Stroke and Fill color”](#) on page 130

[“Work with imported bitmaps”](#) on page 86

Transform gradient and bitmap fills

You can transform a gradient or bitmap fill by adjusting the size, direction, or center of the fill.

- 1 Select the Gradient Transform tool  from the Tools panel. If you do not see the Gradient Transform tool in the Tools panel, click and hold on the Free Transform tool and then select the Gradient Transform tool from the menu that appears.
- 2 Click an area filled with a gradient or bitmap fill. A bounding box with editing handles appears. When the pointer is over any one of these handles, it changes to indicate the function of the handle.

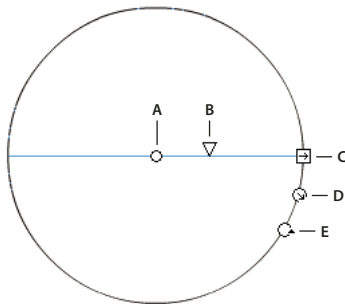
Center point The rollover icon for the center point handle is a four-way arrow.

Focal point The focal point handle appears only when you select a radial gradient. The rollover icon for the focal point handle is an inverted triangle.

Size The rollover icon for the size handle (middle handle icon on the edge of the bounding box) is a circle with an arrow inside of it.

Rotation Adjusts the rotation of the gradient. The rollover icon for the rotation handle (the bottom handle icon on the edge of the bounding box) is four arrows in the shape of a circle.

Width Adjusts the width of the gradient. The rollover icon for the width handle (the square handle) is a double-ended arrow.



Radial gradient controls

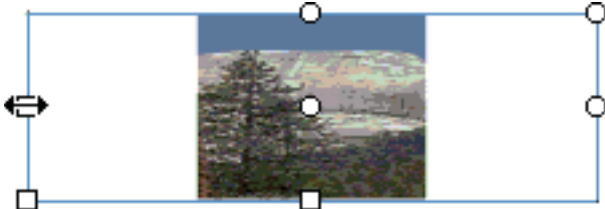
A. Center point B. Width C. Rotation D. Size E. Focal point.

Press Shift to constrain the direction of a linear gradient fill to multiples of 45°.

- 3 Reshape the gradient or fill in any of the following ways:
 - To reposition the center point of the gradient or bitmap fill, drag the center point.



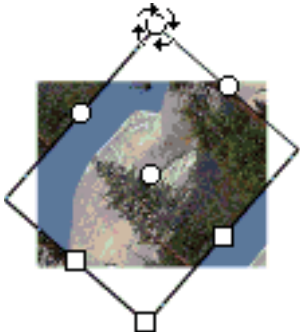
- To change the width of the gradient or bitmap fill, drag the square handle on the side of the bounding box. (This option resizes only the fill, not the object containing the fill.)



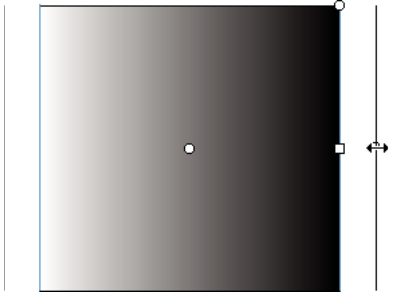
- To change the height of the gradient or bitmap fill, drag the square handle at the bottom of the bounding box.



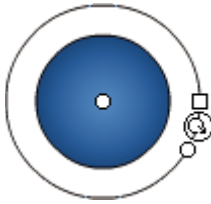
- To rotate the gradient or bitmap fill, drag the circular rotation handle at the corner. You can also drag the lowest handle on the bounding circle of a circular gradient or fill.



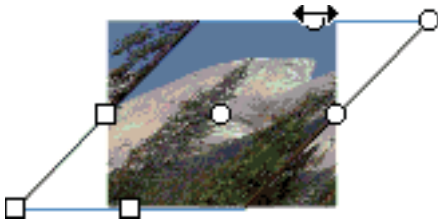
- To scale a linear gradient or a fill, drag the square handle at the center of the bounding box.



- To change the focal point of a circular gradient, drag the middle circular handle on the bounding circle.



- To skew or slant a fill within a shape, drag one of the circular handles on the top or right side of the bounding box.



- To tile a bitmap inside a shape, scale the fill.

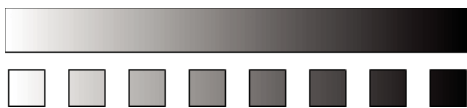


Note: To see all the handles when working with large fills or fills close to the edge of the Stage, select View > Pasteboard.

Lock a gradient or bitmap to fill the Stage

You can lock a gradient or bitmap fill to make it appear that the fill extends over the entire Stage and that the objects painted with the fill are masks revealing the underlying gradient or bitmap.

When you select the Lock Fill modifier with the Brush or Paint Bucket tool and paint with the tool, the bitmap or gradient fill extends across the objects you paint on the Stage.




Using the Lock Fill modifier creates the appearance of a single gradient or bitmap fill being applied to separate objects on the Stage.

See also


[“Modifying painted areas”](#) on page 133

Use a locked gradient fill

- 1 Select the Brush or Paint Bucket tool and select a gradient or bitmap as a fill.
- 2 Select Linear or Radial from the Type menu in the Color panel.
- 3 Click the Lock Fill modifier .

- 4 First paint the areas where you want to place the center of the fill, and then move to other areas.

Use a locked bitmap fill

- 1 Select the bitmap to use.
- 2 Select Bitmap from the Type menu in the Color panel.
- 3 Select the Brush or Paint Bucket tool.
- 4 Click the Lock Fill modifier .
- 5 First paint the areas where you want to place the center of the fill, and then move to other areas.

Kuler panel

About the Kuler panel

The Kuler™ panel is your portal to groups of colors, or themes, created by an online community of designers. You can use it to browse thousands of themes on Kuler™, and then download some to edit or to include in your own projects. You can also use the Kuler panel to create and save themes, and then share them with the Kuler community by uploading them.

The Kuler panel is available in Adobe Photoshop® CS4, Adobe Flash® Professional CS4, Adobe InDesign® CS4, Adobe Illustrator® CS4, and Adobe Fireworks® CS4. The panel is not available in the French versions of these products.

For a video on the Kuler panel, see www.adobe.com/go/lrvid4088_xp.

For an article on Kuler and color inspiration, see Veerle Pieters' blog at http://veerle.duoh.com/blog/comments/adobe_kuler_update_and_color_tips/.

Browse themes

An Internet connection is required to browse themes online.

Search themes

- 1 Select Window > Extensions > Kuler, and then select the Browse panel.
- 2 Do either of the following:
 - In the Search box, enter the name of a theme, tag, or creator.
Note: Use only alphanumerical characters (Aa-Zz, 0-9) in searches.
 - Filter the search results by selecting an option from the pop-up menus above the results.

View a theme online on Kuler

- 1 In the Browse panel, select a theme in the search results.
- 2 Click the triangle on the right side of the theme and select View Online in Kuler.

Saving frequent searches

- 1 Select the Custom option in the first pop-up menu in the Browse panel.
- 2 In the dialog box that opens, enter your search terms and save them.

When you want to run the search, select it from the first pop-up menu.

To delete a saved search, select the Custom option in the pop-up menu. Then clear the searches you want to delete, and click Save.

Working with themes

You can use the Kuler panel to create or edit themes, and include them in your projects.

Note: In Illustrator, you create and edit themes with the Edit Color/Recolor Artwork dialog box, rather than the Create panel. For details, see Illustrator Help.

Add a theme to the Swatches panel of your application

- 1 In the Browse panel, select a theme you want to use.
- 2 Click the triangle on the right side of the theme and select Add To Swatches Panel.

You can also add a theme from the Create panel by clicking the Add To Swatches button at the bottom of the panel.

Edit a theme

- 1 In the Browse panel, locate a theme you want to edit and then double-click the theme in the search results. The theme opens in the Create panel.
- 2 In the Create panel, edit the theme using the tools at your disposal. For more information, see the Create Panel Tools topic that follows.
- 3 Do one of the following:
 - Save your theme by clicking the Save Theme button.
 - Add the theme to your application's Swatches panel by clicking the Add To Swatches Panel button at the bottom of the panel.
 - Upload the theme to the Kuler service by clicking the Upload button at the bottom of the panel.

Create panel tools

The Create panel provides a variety of tools to create or edit themes.

- Select a harmony rule from the Select Rule pop-up menu. The harmony rule uses the base color as the basis for generating the colors in the color group. For example, if you choose a blue base color and the Complementary harmony rule, a color group is created using the base color, blue, and its complement, red.
- Select the Custom rule to create a theme using free-form adjustments.
- Manipulate the colors in the color wheel. As you make your adjustments, the selected harmony rule continues to govern the colors generated for the color group.
- Move the Brightness slider beside the wheel to adjust the color brightness.
- Set the base color by dragging the Base color marker (the largest, double-ringed color marker) around the wheel. You can also set the base color by adjusting the color sliders at the bottom of the dialog box.
- Set one of the four other colors in the color group as the base color. Select the color's swatch and click the bull's-eye button below the color group.
- Set the host application's foreground/background color or stroke/fill color as the base color. Click one of the first two buttons below the color group.
- Remove a color from the color group by selecting the color's swatch and clicking the Remove Color button below the color group. Add a new color by selecting an empty color swatch and clicking the Add Color button.
- Try different color effects by selecting a new harmony rule and by moving the markers in the color wheel.

- Double-click any of the swatches in the color group to set the active color (foreground/background or stroke/fill) in your application. If the application doesn't have an active or selected color feature, the Kuler panel sets the foreground color or the fill color as appropriate.

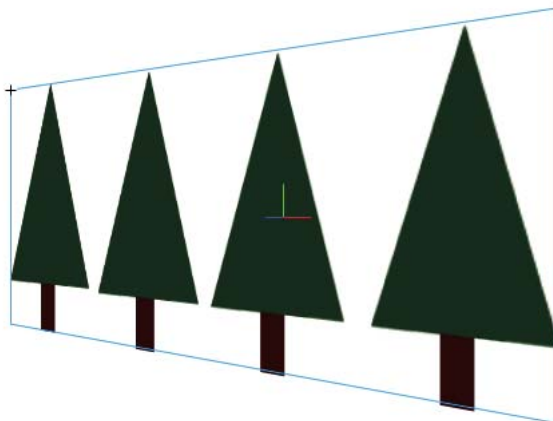
3D graphics

About 3D graphics in Flash

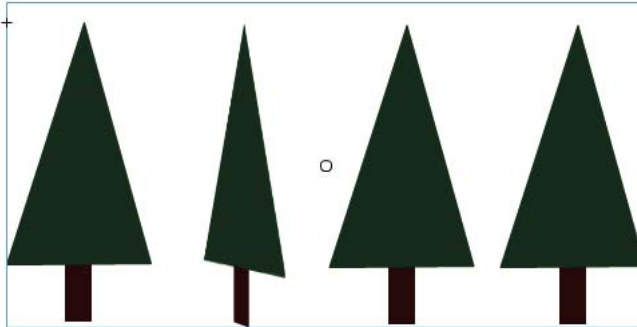
Flash allows you to create 3D effects by moving and rotating movie clips in 3D space on the Stage. Flash represents 3D space by including a z axis in the properties of each movie clip instance. You add 3D perspective effects to movie clip instances by moving or rotating them along their z axis using the 3D Translation and 3D Rotation tools. In 3D terminology, moving an object in 3D space is called a *translation* and rotating an object in 3D space is called a *transformation*. Once you have applied either of these effects to a movie clip, Flash considers it a 3D movie clip and a colored axes indicator appears overlaid onto the movie clip whenever it is selected.

To make an object appear nearer or further away from the viewer, move it along its z axis with the 3D Translation tool or the Property inspector. To give the impression of an object that is at an angle to the viewer, rotate the movie clip around its z axis with the 3D Rotation tool. By using these tools in combination, you can create realistic perspective effects.

Both the 3D Translation and the 3D Rotation tools allow you to manipulate objects in global or local 3D space. Global 3D space is the Stage space. Global transforms and translations are relative to the Stage. Local 3D space is the movie clip space. Local transforms and translations are relative to the movie clip space. For example, if you have a movie clip containing several nested movie clips, local 3D transforms of the nested movie clips are relative to the drawing area inside the container movie clip. The default mode of the 3D Translation and Rotation tools is global. To use them in local mode, click the Global toggle button in the Options section of the Tools panel.



The Stage with a movie clip rotated in global 3D space.



The Stage with a movie clip containing a nested movie clip rotated in local 3D space.

By using the 3D properties of movie clip instances in your FLA file, you can create a variety of graphic effects without duplicating movie clips in the library. However, when you edit a movie clip from the library, 3D transforms and translations that have been applied are not visible. When editing the contents of a movie clip, only 3D transforms of nested movie clips are visible.

Note: Once a 3D transform is added to a movie clip instance, its parent movie clip symbol cannot be edited in Edit in Place mode.

If you have 3D objects on the Stage, you can add certain 3D effects to all of those objects as a group by adjusting the Perspective Angle and Vanishing Point properties of your FLA file. The Perspective Angle property has the effect of zooming the view of the Stage. The Vanishing Point property has the effect of panning the 3D objects on the Stage. These settings only affect the appearance of movie clips that have a 3D transform or translation applied to them.


In the Flash authoring tool, you can control only one viewpoint, or *camera*. The camera view of your FLA file is the same as the Stage view. Each FLA file has only one perspective Angle and Vanishing Point setting.

To use the 3D capabilities of Flash, the publish settings of your FLA file must be set to Flash Player 10 and ActionScript 3.0. Only movie clip instances can be rotated or translated along the z axis. Some 3D capabilities are available through ActionScript that are not available directly in the Flash user interface, such as multiple Vanishing Points and separate cameras for each movie clip. Using ActionScript 3.0, you can apply 3D properties to objects such as text, FLV Playback components, and buttons, in addition to movie clips.

Note: The 3D tools cannot be used on objects on mask layers and layers containing 3D objects cannot be used as mask layers. For more information about mask layers, see “Using mask layers” on page 221.

For a video tutorial about 3D graphics, see Working with 3D Art at www.adobe.com/go/lrvid4059_fl.

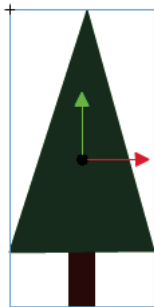
Move objects in 3D space

You move movie clip instances in 3D space with the 3D Translation tool . When you select a movie clip with the tool, its three axes, X, Y, and Z, appear on the Stage on top of the object. The x axis is red, the y axis is green, and the z axis is blue.

The default mode of the 3D Translation tool is global. Moving an object in global 3D space is the same as moving it relative to the Stage. Moving an object in local 3D space is the same as moving it relative to its parent movie clip if it has one. To toggle the 3D Translation tool between global and local modes, click the Global toggle button in the Options section of the Tools panel while the 3D Translation tool is selected. You can temporarily toggle the mode from global to local by pressing the D key while dragging with the 3D Translation tool.

The 3D Translation and Rotation tools occupy the same space in the Tools panel. Click and hold the active 3D tool icon in the Tools panel to select the currently inactive 3D tool.


By default, selected objects that have 3D translation applied appear with a 3D axis overlay on the Stage. You can turn off this overlay in the General section of Flash Preferences.




The 3D Translation tool overlay.

Note: Changing the z axis position of a 3D movie clip will cause the movie clip to appear to change its x and y position as well. This is because movement along the z axis follows the invisible perspective lines which radiate from the 3D vanishing point (set in the 3D symbol instance Property inspector) to the edges of the Stage.

Move a single object in 3D space

- 1 Select the 3D Translation tool  in the Tools panel (or press the G key to select it).
- 2 Set the tool to Local or Global mode.

Be sure that the tool is in the mode you want mode by checking the Global toggle button in the Options section of the Tools panel. Click the button or press the D key to toggle the mode.


- 3 Select a movie clip with the 3D Translation tool .
- 4 To move the object by dragging with the tool, move the pointer over the x-, y-, or z-axis controls. The pointer changes when over any of the controls.

The x- and y-axis controls are the arrow tips on each axis. Drag one of these controls in the direction of its arrow to move the object along the selected axis. The z-axis control is the black dot at the center of the movie clip. Drag the z-axis control up or down to move the object on the z axis.

- 5 To move the object using the Property inspector, enter a value for X, Y, or Z in the 3D Position and View section of the Property inspector.

When you move an object on the z axis, its apparent size changes. The apparent size appears in the Property inspector as the Width and Height values in the 3D Position and View section of the Property inspector. These values are read-only.


Move a multiple-object selection in 3D space

When you select multiple movie clips, you can move one of the selected objects with the 3D Translation tool  and the others move in the same way.

- To move each object in the group in the same way in global 3D space, set the 3D Translation tool to global mode and then drag one of the objects with the axis controls. Shift-double-click one of the selected objects to move the axis controls to that object.
- To move each object in the group in the same way in local 3D space, set the 3D Translation tool to local mode and then drag one of the objects with the axis controls. Shift-double-click one of the selected objects to move the axis controls to that object.

You can also move the axis controls to the center of the multiple selection by double-clicking the z-axis control. Shift-double-click one of the selected objects to move the axis controls to that object.

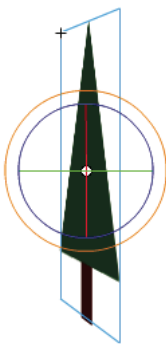
Rotate objects in 3D space

You rotate movie clip instances in 3D space with the 3D Rotation tool . A 3D rotation control appears on top of selected objects on the Stage. The X control is red, the Y control is green, and the Z control is blue. Use the orange free rotate control to rotate around the X and Y axes at the same time.

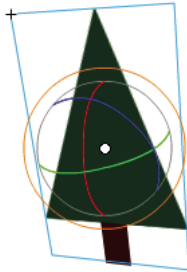
The default mode of the 3D Rotation tool is global. Rotating an object in global 3D space is the same as moving it relative to the Stage. Rotating an object in local 3D space is the same as moving it relative to its parent movie clip if it has one. To toggle the 3D Rotation tool between global and local modes, click the Global toggle button in the Options section of the Tools panel while the 3D Rotation tool is selected. You can temporarily toggle the mode from global to local by pressing the D key while dragging with the 3D Rotation tool.

The 3D Rotation and Translation tools occupy the same space in the Tools panel. Click and hold the active 3D tool icon in the Tools panel to select the currently inactive 3D tool.

By default, selected objects that have 3D rotation applied appear with a 3D axis overlay on the Stage. You can turn off this overlay in the General section of Flash Preferences.




The Global 3D Rotation tool overlay.



The Local 3D Rotation tool overlay.

Rotate a single object in 3D space

- 1 Select the 3D Rotation tool  in the Tools panel (or press the W key).

Verify that the tool is in the mode that you want by checking the Global toggle button in the Options section of the Tools panel. Click the button or press the D key to toggle the mode between global and local.

- 2 Select a movie clip on the Stage.

The 3D Rotation controls appear overlaid on selected object. If the controls appear in a different location, double-click the control center point to move it to the selected object.

- 3 Place the pointer over one of the four rotation axis controls.

The pointer changes when over one of the four controls.

- 4 Drag one of the axis controls to rotate around that axis, or the free rotate control (outer orange circle) to rotate x and y simultaneously.


Drag the X axis control left or right to rotate around the x axis. Drag the y-axis control up or down to rotate around the y axis. Drag the z-axis control in a circular motion to rotate around the z axis.

- 5 To relocate the rotation control center point relative to the movie clip, drag the center point. To constrain the movement of the center point to 45° increments, press the Shift key while dragging.

Moving the rotation control center point lets you control the effect of the rotation on the object and its appearance. Double-click the center point to move it back to the center of the selected movie clip.

The location of the rotation control center point for the selected object appears in the Transform panel as the 3D Center Point property. You can modify the location of the center point in the Transform panel.

Rotate a multiple-object selection in 3D space

- 1 Select the 3D Rotation tool  in the Tools panel (or press the W key).

Verify that the tool is in the mode that you want by checking the Global toggle button in the Options section of the Tools panel. Click the button or press the D key to toggle the mode between global and local.

- 2 Select multiple movie clips on the Stage.

The 3D Rotation controls appear overlaid on the most recently selected object.

- 3 Place the pointer over one of the four rotation axis controls.

The pointer changes when over one of the four controls.

- 4 Drag one of the axis controls to rotate around that axis, or the free rotate control (outer orange circle) to rotate X and Y simultaneously.

Drag the x-axis control left or right to rotate around the x axis. Drag the y-axis control up or down to rotate around the y axis. Drag the z-axis control in a circular motion to rotate around the z axis.

All of the selected movie clips rotate around the 3D center point which appears at the center of the rotation controls.

- 5 To relocate the 3D rotation control center point, do one of the following:

- To move the center point to an arbitrary location, drag the center point.
- To move the center point to the center of one of the selected movie clips, Shift-double-click the movie clip.
- To move the center point to the center of the group of selected movie clips, double-click the center point.

Changing the location of the 3D rotation center point lets you control the effect of the rotation on the objects.

The location of the rotation control center point for the selected object appears in the Transform panel as the 3D Center Point. You can modify the location of the center point in the Transform panel.

Rotate a selection with the Transform panel

- 1 Open the Transform panel (Window > Transform).
- 2 Select one or more movie clips on the Stage.
- 3 In the Transform panel, enter the desired values in the 3D Rotation X, Y, and Z fields to rotate the selection. These fields contain hot text, so you can drag the values to change them.

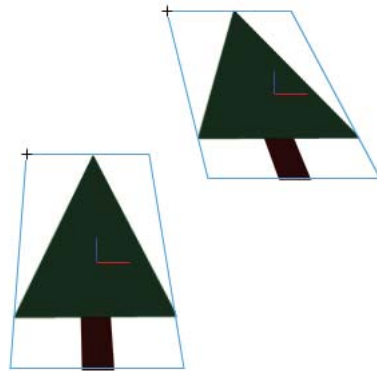
Note: The 3D rotation takes place in global or local 3D space, depending on the current mode of the 3D Rotation tool in the Tools panel.

- 4 To move the 3D rotation point, enter the desired values in the 3D Center Point X, Y, and Z fields.

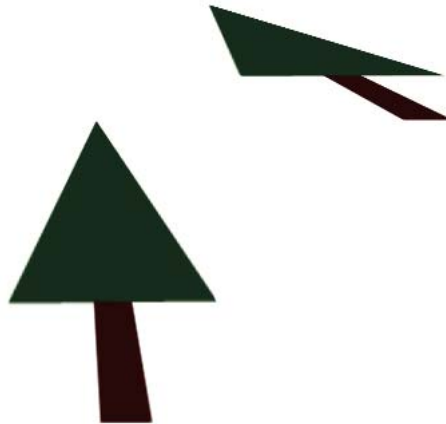
Adjusting the Perspective Angle

The Perspective Angle property of a FLA file controls the apparent angle of view of 3D movie clips on the Stage.

Increasing or decreasing the Perspective Angle affects the apparent size of 3D movie clips and their location relative to the edges of the Stage. By increasing the Perspective Angle, you make 3D objects appear closer to the viewer. By decreasing the Perspective Angle property, you make 3D objects appear further away. The effect is like zooming in or out with a camera lens which changes the angle of view through the lens.



The Stage with a Perspective Angle of 55.



The Stage with a Perspective Angle of 110.

The Perspective Angle property affects all movie clips with 3D translation or rotation applied to them. The Perspective Angle does not affect other movie clips. The default Perspective Angle is 55° of view, like a normal camera lens. The range of values is from 1° to 180°.

To view or set the Perspective Angle in the Property inspector, a 3D movie clip must be selected on Stage. Changes to the Perspective Angle are visible on the Stage immediately.

The Perspective Angle changes automatically when you change the Stage size so that the appearance of 3D objects does not change. You can turn off this behavior in the Document Properties dialog box.

To set the Perspective Angle:

- 1 On the Stage, select a movie clip instance that has 3D rotation or translation applied to it.
- 2 In the Property inspector, enter a new value in the Perspective Angle field, or drag the hot text to change the value.

Adjust the Vanishing Point

The Vanishing Point property of a FLA file controls the orientation of the z axis of 3D movie clips on the Stage. The z axes of all 3D movie clips in a FLA file recede toward the Vanishing Point. By relocating the Vanishing Point, you change the direction that an object moves when translated along its z axis. By adjusting the position of the Vanishing Point, you can precisely control the appearance of 3D objects and animation on the Stage.

For example, if you locate the Vanishing Point at the upper-left corner of the Stage (0, 0), then increasing the value of the Z property of a movie clip moves the movie clip away from the viewer and toward the upper-left corner of the Stage.

Because the Vanishing Point affects all 3D movie clips, changing it also changes the position of all movie clips that have a z-axis translation applied.

The Vanishing Point is a document property that affects all movie clips that have z-axis translation or rotation applied to them. The Vanishing Point does not affect other movie clips. The default location of the Vanishing Point is the center of the Stage.

To view or set the Vanishing Point in the Property inspector, a 3D movie clip must be selected on Stage. Changes to the Vanishing Point are visible on the Stage immediately.

To set the Vanishing Point:

- 1 On the Stage, select a movie clip that has 3D rotation or translation applied to it.
- 2 In the Property inspector, enter a new value in the Vanishing Point field, or drag the hot text to change the value. Guides indicating the location of the Vanishing Point appear on the Stage while dragging the hot text.
- 3 To move the Vanishing Point back to the center of the Stage, click the Reset button in the Property inspector.

Chapter 6: Symbols, instances, and library assets

Adobe® Flash® CS4 Professional allows you to import and create many kinds of assets to populate your Flash documents. These assets are managed in Flash as *symbols*, *instances*, and *library assets*. Understanding how these types of assets work together lets you make good choices about how and when to use them, and anticipate the best design options for your work.

Working with symbols

About symbols

A *symbol* is a graphic, button, or movie clip that you create once in the Flash authoring environment or by using the Button (AS 2.0), SimpleButton (AS 3.0), and MovieClip classes. You can then reuse the symbol throughout your document or in other documents.

A symbol can include artwork that you import from another application. Any symbol that you create automatically becomes part of the library for the current document.

An *instance* is a copy of a symbol located on the Stage or nested inside another symbol. An instance can be different from its symbol in color, size, and function. Editing the symbol updates all of its instances, but applying effects to an instance of a symbol updates only that instance.

Using symbols in your documents dramatically reduces file size; saving several instances of a symbol requires less storage space than saving multiple copies of the contents of the symbol. For example, you can reduce the file size of your documents by converting static graphics, such as background images, into symbols and then reusing them. Using symbols can also speed SWF file playback, because a symbol needs to be downloaded to Flash® Player only once.

Share symbols among documents as shared library assets during authoring or at runtime. For runtime shared assets, you can link assets in a source document to any number of destination documents, without importing the assets into the destination document. For assets shared during authoring, you can update or replace a symbol with any other symbol available on your local network.




If you import library assets with the same name as assets already in the library, you can resolve naming conflicts without accidentally overwriting existing assets.

Additional introductory instruction about symbols is available from these resources:

- Adobe TV: [Understanding symbols](#)
- Adobe Video Workshop: [Creating and using symbols and instances](#) (This video focuses on Flash Professional CS3, but the information is still relevant.)
- Flash Professional Design Center article: [Using Flash for the first time – Part 1: Building a banner](#)

Types of symbols

Each symbol has a unique Timeline and Stage, complete with layers. You can add frames, keyframes, and layers to a symbol Timeline, just as you can to the main Timeline. When you create a symbol you choose the symbol type.

- Use graphic symbols  for static images and to create reusable pieces of animation that are tied to the main Timeline. Graphic symbols operate in sync with the main Timeline. Interactive controls and sounds won't work in a graphic symbol's animation sequence. Graphic symbols add less to the FLA file size than buttons or movie clips because they have no timeline.
- Use button symbols  to create interactive buttons that respond to mouse clicks, rollovers, or other actions. You define the graphics associated with various button states, and then assign actions to a button instance. For more information, see Handling events in [Learning ActionScript 2.0 in Adobe Flash](#) or Handling events in *Programming ActionScript 3.0*.
- Use movie clip symbols  to create reusable pieces of animation. Movie clips have their own multiframe Timeline that is independent from the main Timeline—think of them as nested inside a main Timeline that can contain interactive controls, sounds, and even other movie clip instances. You can also place movie clip instances inside the Timeline of a button symbol to create animated buttons. In addition, movie clips are scriptable with ActionScript®.
- Use font symbols to export a font and use it in other Flash documents.

Flash provides built-in *components*, movie clips with defined parameters, that you can use to add user interface elements, such as buttons, checkboxes, or scroll bars, to your documents. For more information, see About Components in *Using ActionScript 2.0 Components* at http://www.adobe.com/go/learn_fl_cs4_as2components_en, or [About ActionScript 3.0 Components](#) in *Using ActionScript 3.0 Components*.

Note: To preview animation in component instances and scaling of 9-slice-scaled movie clips in the Flash authoring environment, select Control > Enable Live Preview.

See also

“[Working with button symbols](#)” on page 164

“[Sharing library assets](#)” on page 162

“[Managing assets with the library](#)” on page 157

“[Embed and share fonts](#)” on page 251

Create symbols

You can create a symbol from selected objects on the Stage, create an empty symbol and make or import the content in symbol-editing mode, and create font symbols in Flash. Symbols can contain all the functionality that Flash can create, including animation.

Using symbols that contain animation lets you create Flash applications with a lot of movement while minimizing file size. Consider creating animation in a symbol that has a repetitive or cyclic action—the up-and-down motion of a bird's wings, for example.

To add symbols to your document, use shared library assets during authoring or at runtime.

See also

[“Editing instance properties”](#) on page 153

[“Sharing library assets”](#) on page 162

[“Embed and share fonts”](#) on page 251

Convert selected elements to a symbol

- 1 Select an element or several elements on the Stage. Do one of the following:
 - Select Modify > Convert To Symbol.
 - Drag the selection to the Library panel.
 - Right-click (Windows) or Control-click (Macintosh) and select Convert To Symbol from the context menu.
- 2 In the Convert To Symbol dialog box, type the name of the symbol and select the behavior.
- 3 Click in the registration grid to position the registration point for the symbol.
- 4 Click OK.

Flash adds the symbol to the library. The selection on the Stage becomes an instance of the symbol. Once you have created a symbol, you can edit it in symbol edit mode by choosing Edit > Edit Symbols, or you can edit it in the context of the Stage by choosing Edit > Edit In Place. You can also change the registration point of a symbol.

Create an empty symbol

- 1 Do one of the following:
 - Select Insert > New Symbol.
 - Click the New Symbol button at the lower left of the Library panel.
 - Select New Symbol from the Library Panel menu in the upper-right corner of the Library panel.
- 2 In the Create New Symbol dialog box, type the name of the symbol and select the behavior.
- 3 Click OK.

Flash adds the symbol to the library and switches to symbol-editing mode. In symbol-editing mode, the name of the symbol appears above the upper-left corner of the Stage, and a cross hair indicates the symbol’s registration point.

- 4 To create the symbol content, use the Timeline, draw with the drawing tools, import media, or create instances of other symbols.
- 5 To return to document-editing mode, do one of the following:
 - Click the Back button.
 - Select Edit > Edit Document.
 - Click the scene name in the Edit bar.

When you create a symbol, the registration point is placed at the center of the window in symbol-editing mode. You can place the symbol contents in the window in relation to the registration point. To change the registration point, when you edit a symbol, move the symbol contents in relation to the registration point.

Convert animation on the Stage into a movie clip symbol

To reuse an animated sequence on the Stage, or to manipulate it as an instance, select it and save it as a movie clip symbol.

- 1 On the main Timeline, select every frame in every layer of the animation on the Stage that you want to use. For information on selecting frames, see “[Insert frames in the Timeline](#)” on page 172.
- 2 Do one of the following to copy the frames:
 - Right-click (Windows) or Control-click (Macintosh) any selected frame, and select Copy Frames from the context menu. To delete the sequence after converting it to a movie clip, select Cut.
 - Select Edit > Timeline > Copy Frames. To delete the sequence after converting it to a movie clip, select Cut Frames.
- 3 Deselect your selection and make sure nothing on the Stage is selected. Select Insert > New Symbol.
- 4 Name the symbol. For Type, select Movie Clip, then click OK.
- 5 On the Timeline, click Frame 1 on Layer 1, and select Edit > Timeline > Paste Frames.

This action pastes the frames (and any layers and layer names) you copied from the main Timeline to the Timeline of this movie clip symbol. Any animation, buttons, or interactivity from the frames you copied now becomes an independent animation (a movie clip symbol) that you can reuse.

- 6 To return to document-editing mode, do one of the following:
 - Click the Back button.
 - Select Edit > Edit Document.
 - Click the scene name in the Edit bar above the Stage.

Duplicate symbols

Duplicating a symbol lets you use an existing symbol as a starting point for creating a symbol.

To create versions of the symbol with different appearances, also use instances.

Duplicate a symbol using the Library panel

- ❖ Select a symbol in the Library panel and do one of the following:
 - Right-click (Windows) or Control-click (Macintosh), and select Duplicate from the context menu.
 - Select Duplicate from the Library Panel menu.

Duplicate a symbol by selecting an instance

- 1 Select an instance of the symbol on the Stage.
- 2 Select Modify > Symbol > Duplicate Symbol.

The symbol is duplicated, and the instance is replaced with an instance of the duplicate symbol.

Edit symbols

When you edit a symbol, Flash updates all the instances of that symbol in your document. Edit the symbol in the following ways:

- In context with the other objects on the Stage by using the Edit In Place command. Other objects are dimmed to distinguish them from the symbol you are editing. The name of the symbol you are editing appears in an Edit bar at the top of the Stage, to the right of the current scene name.

- In a separate window, using the Edit In New Window command. Editing a symbol in a separate window lets you see the symbol and the main Timeline at the same time. The name of the symbol you are editing appears in the Edit bar at the top of the Stage.

You edit the symbol by changing the window from the Stage view to a view of only the symbol, using symbol-editing mode. The name of the symbol you are editing appears in the Edit bar at the top of the Stage, to the right of the current scene name.

When you edit a symbol, Flash updates all instances of the symbol throughout the document to reflect your edits. While editing a symbol, use any of the drawing tools, import media, or create instances of other symbols.

- Change the registration point of a symbol (the point identified by the coordinates 0, 0) by using any symbol-editing method.

Edit a symbol in place

1 Do one of the following:

- Double-click an instance of the symbol on the Stage.
- Select an instance of the symbol on the Stage and right-click (Windows) or Control-click (Macintosh), and select Edit in Place.
- Select an instance of the symbol on the Stage, and select Edit > Edit In Place.

2 Edit the symbol.

3 To change the registration point, drag the symbol on the Stage. A cross hair indicates the location of the registration point.

4 To exit edit-in-place mode and return to document-editing mode, do one of the following:

- Click the Back button.
- Select the current scene name from the Scene menu in the Edit bar.
- Select Edit > Edit Document.
- Double-click outside the symbol content.

Edit a symbol in a new window

1 Select an instance of the symbol on the Stage and right-click (Windows) or Control-click (Macintosh), and select Edit In New Window.

2 Edit the symbol.

3 To change the registration point, drag the symbol on the Stage. A cross hair indicates the location of the registration point.

4 Click the Close box in the upper-right corner (Windows) or upper-left corner (Macintosh) to close the new window, and click in the main document window to return to editing the main document.

Edit a symbol in symbol-editing mode

1 Do one of the following to select the symbol:

- Double-click the symbol's icon in the Library panel.
- Select an instance of the symbol on the Stage, and right-click (Windows) or Control-click (Macintosh), and select Edit from the context menu.
- Select an instance of the symbol on the Stage and select Edit > Edit Symbols.

- Select the symbol in the Library panel and select Edit from the Library Panel menu, or right-click (Windows) or Control-click (Macintosh) the symbol in the Library panel and select Edit.
- 2 Edit the symbol.
 - 3 To exit symbol-editing mode and return to editing the document, do one of the following:
 - Click the Back button at the left of the Edit bar at the top of the Stage.
 - Select Edit > Edit Document.
 - Click the scene name in the Edit bar at the top of the Stage.
 - Double-click outside the symbol content.

Working with symbol instances

Create instances

After you create a symbol, you can create instances of that symbol throughout your document, including inside other symbols. When you modify the symbol, Flash updates all instances of the symbol.

You can give names to instances from the Property inspector. Use the instance name to refer to an instance in ActionScript. To control instances with ActionScript®, give each instance within a single timeline a unique name. For more information, see Handling events in [Learning ActionScript 2.0 in Adobe Flash](#) or Handling events in *Programming ActionScript 3.0*.

To specify color effects, assign actions, set the graphic display mode, or change the behavior of new instances, use the Property inspector. The behavior of the instance is the same as the symbol behavior, unless you specify otherwise. Any changes you make affect only the instance and not the symbol.

Create an instance of a symbol

- 1 Select a layer in the Timeline. Flash can place instances only in keyframes, always on the current layer. If you don't select a keyframe, Flash adds the instance to the first keyframe to the left of the current frame.

Note: A keyframe is a frame in which you define a change in the animation. For more information, see [“Insert frames in the Timeline”](#) on page 172.

- 2 Select Window > Library.
- 3 Drag the symbol from the library to the Stage.
- 4 If you created an instance of a graphic symbol, to add the number of frames that will contain the graphic symbol, select Insert > Timeline > Frame.

Apply a custom name to an instance

- 1 Select the instance on the Stage.
- 2 Select Window > Properties, and enter a name in the Instance Name box.

Editing instance properties

Each symbol instance has its own properties that are separate from the symbol. You can change the tint, transparency, and brightness of an instance; redefine how the instance behaves (for example, change a graphic to a movie clip); and specify how an animation plays inside a graphic instance. You can also skew, rotate, or scale an instance without affecting the symbol.

In addition, you can name a movie clip or button instance so that you can use ActionScript to change its properties. For more information, see *Classes* in *Learning ActionScript 2.0 in Adobe Flash* at http://www.adobe.com/go/learn_fl_cs4_learningAS2_en or *Objects and classes* in *Programming ActionScript 3.0*. To edit instance properties, you use the Property inspector (Windows > Properties).

The properties of an instance are saved with it. If you edit a symbol or relink an instance to a different symbol, any instance properties you've changed still apply to the instance.

Change the color and transparency of an instance

Each instance of a symbol can have its own color effect. To set color and transparency options for instances, use the Property inspector. Settings in the Property inspector also affect bitmaps placed in symbols.

When you change the color and transparency for an instance in a specific frame, Flash makes the change as soon as it displays that frame. To make gradual color changes, apply a motion tween. When tweening color, you enter different effect settings in starting and ending keyframes of an instance, and then tween the settings to make the instance's colors shift over time.



Tweening gradually changes an instance's color or transparency.

Note: If you apply a color effect to a movie clip symbol that has multiple frames, Flash applies the effect to every frame in the movie clip symbol.

- 1 Select the instance on the Stage, and select Window > Properties.

2 In the Property inspector, select one of the following options from the Style menu in the Color Effect section:

Brightness Adjusts the relative lightness or darkness of the image, measured on a scale from black (–100%) to white (100%). To adjust brightness, click the triangle and drag the slider or enter a value in the box.

Tint Colors the instance with the same hue. To set the tint percentage from transparent (0%) to completely saturated (100%), use the Tint slider in the Property inspector. To adjust tint, click the triangle and drag the slider or enter a value in the box. To select a color, enter red, green, and blue values in the respective boxes, or click the Color control and select a color from the Color Picker.

Alpha Adjusts the transparency of the instance, from transparent (0%) to completely saturated (100%). To adjust the alpha value, click the triangle and drag the slider or enter a value in the box.

Advanced Separately adjusts the red, green, blue, and transparency values of an instance. This is most useful to create and animate subtle color effects on objects such as bitmaps. The controls on the left let you reduce the color or transparency values by a specified percentage. The controls on the right let you reduce or increase the color or transparency values by a constant value.

The current red, green, blue, and alpha values are multiplied by the percentage values, and then added to the constant values in the right column, producing the new color values. For example, if the current red value is 100, setting the left slider to 50% and the right slider to 100% produces a new red value of 150 ($[100 \times .5] + 100 = 150$).

***Note:** The Advanced settings in the Effect panel implement the function $(a * y + b) = x$ where a is the percentage specified in the left set of boxes, y is the color of the original bitmap, b is the value specified in the right set of boxes, and x is the resulting effect (between 0 and 255 for RGB, and 0 and 100 for alpha transparency).*

You can also change the color of an instance using the ActionScript ColorTransform object. For detailed information on the Color object, see ColorTransform in *ActionScript 2.0 Language Reference* or *ActionScript 3.0 Language and Components Reference*.

See also

[“Add a classic tween to instances, groups, or type”](#) on page 212

Swap one instance for another

To display a different instance on the Stage and preserve all the original instance properties, such as color effects or button actions, assign a different symbol to an instance.

For example, suppose you’re creating a cartoon with a rat symbol for your character, but decide to change the character to a cat. You could replace the rat symbol with the cat symbol and have the updated character appear in roughly the same location in all your frames.

See also

[“Working with button symbols”](#) on page 164

Assign a different symbol to an instance

- 1 Select the instance on the Stage, and select Window > Properties.
- 2 Click the Swap button in the Property inspector.
- 3 Select a symbol to replace the symbol currently assigned to the instance. To duplicate a selected symbol, click Duplicate Symbol and click OK.

Duplicating lets you base a new symbol on an existing one in the library and minimizes copying if you’re making several symbols that differ slightly.

Replace all instances of a symbol

- ❖ Drag a symbol with the same name as the symbol you are replacing from one Library panel into the Library panel of the FLA file you are editing and click Replace. If you have folders in the library, the new symbol must be dragged into the same folder as the symbol you are replacing.

Change an instance's type

To redefine an instance's behavior in a Flash application, change its type. For example, if a graphic instance contains animation that you want to play independently of the main Timeline, redefine the graphic instance as a movie clip instance.

- 1 Select the instance on the Stage, and select Window > Properties.
- 2 Select Graphic, Button, or Movie Clip from the menu in the Property inspector.

Set looping for a graphic instance

To determine how animation sequences inside a graphic instance play in your Flash application, set options in the Property inspector.

An animated graphic symbol is tied to the Timeline of the document in which the symbol is placed. In contrast, a movie clip symbol has its own independent Timeline. Animated graphic symbols, because they use the same Timeline as the main document, display their animation in document-editing mode. Movie clip symbols appear as static objects on the Stage and do not appear as animations in the Flash editing environment.

- 1 Select a graphic instance on the Stage, and select Window > Properties.
- 2 Select an animation option from the Options menu in the Looping section of the Property inspector:

Loop Loops all the animation sequences contained in the current instance for as many frames as the instance occupies.

Play Once Plays the animation sequence beginning from the frame you specify to the end of the animation and then stops.

Single Frame Displays one frame of the animation sequence. Specify which frame to display.

- 3 To specify the first frame of the graphic symbol to display when looping, enter a frame number in the First text box. The Single Frame option also uses the frame number you specify here.

Break apart an instance symbol

To break the link between an instance and a symbol and make the instance into a collection of ungrouped shapes and lines, you *break apart* the instance. This feature is useful for changing the instance substantially without affecting any other instance. If you modify the source symbol after breaking apart the instance, the instance is not updated with the changes.

- 1 Select the instance on the Stage.
- 2 Select Modify > Break Apart. This action breaks the instance into its component graphic elements.
- 3 To modify these elements, use the painting and drawing tools.

Get information about instances on the Stage

The Property inspector and Info panel display the following information about instances selected on the Stage:

- In the Property inspector, view the instance's behavior and settings—for all instance types, color effect settings, location, and size; for graphics, the loop mode and first frame that contains the graphic; for buttons, the instance name (if assigned) and tracking option; for movie clips, the instance name (if assigned). For location, the Property inspector displays the x and y coordinates of either the symbol's registration point or the symbol's upper-left corner, depending on which option is selected in the Info panel.
- In the Info panel, view the instance's size and location; the location of its registration point; its red (R), green (G), blue (B), and alpha (A) values (if the instance has a solid fill); and the location of the pointer. The Info panel also displays the x and y coordinates of either the symbol's registration point or the symbol's upper-left corner, depending on which option is selected. To display the coordinates of the registration point, click the center square in the Coordinate grid in the Info panel. To display the coordinates of the upper-left corner, click the upper-left square in the Coordinate grid.
- In the Movie Explorer, view the contents of the current document, including instances and symbols.
View any actions assigned to a button or movie clip in the Actions panel.

See also

[“Using the Movie Explorer with screens”](#) on page 320

Get information about an instance

- 1 Select the instance on the Stage.
- 2 Display the Property inspector (Window > Properties) or panel to use:
 - To display the Info panel, select Window > Info.
 - To display the Movie Explorer, select Window > Movie Explorer.
 - To display the Actions panel, select Window > Actions.

View the symbol definition for the selected symbol in the Movie Explorer

- 1 Click the Show Buttons, Movie Clips, and Graphics button at the top of the Movie Explorer.
- 2 Right-click (Windows) or Control-click (Macintosh), and select Show Symbol Instances and Go To Symbol Definition; or select these options from the menu in the upper-right corner of the Movie Explorer.

Jump to the scene containing instances of a selected symbol

- 1 Display the symbol definitions.
- 2 Right-click (Windows) or Control-click (Macintosh), and select Show Movie Elements and Go To Symbol Definition; or select these options from the menu in the upper-right corner of the Movie Explorer.

Working with the library

Managing assets with the library

The library in a Flash document stores media assets that you create in the Flash authoring environment or import to use in the document. You can create vector artwork or text directly in Flash; import vector artwork, bitmaps, video, and sound; and create symbols. A *symbol* is a graphic, a button, a movie clip, or text that you create once and can reuse multiple times. You can also use ActionScript to add media content to a document dynamically.

The library also contains any components that you have added to your document. Components appear in the library as compiled clips.

You can open the library of any Flash document while you are working in Flash, to make the library items from that file available for the current document.

You can create permanent libraries in your Flash application that are available whenever you start Flash. Flash also includes several sample libraries containing buttons, graphics, movie clips, and sounds.

You can export library assets as a SWF file to a URL to create a runtime-shared library. This lets you link to the library assets from Flash documents that import symbols using runtime sharing.

See also

[“Text”](#) on page 244

[“Using imported artwork”](#) on page 63

[“Sound”](#) on page 273

[“Video”](#) on page 283

[“Symbols, instances, and library assets”](#) on page 147

Work with libraries

The Library panel (Window > Library) displays a scroll list with the names of all items in the library, which lets you view and organize these elements as you work. An icon next to an item’s name in the Library panel indicates the item’s file type.

Open a library in another Flash file

- 1 From the current document, select File > Import > Open External Library.
- 2 Navigate to the Flash file whose library you want to open and click Open.

The selected file’s library opens in the current document, with the filename at the top of the Library panel. To use items from the selected file’s library in the current document, drag the items to the current document’s Library panel or to the Stage.

Resize the Library panel

- ❖ Do one of the following:
 - Drag the lower-right corner of the panel.
 - Click the Wide State button to enlarge the Library panel so it shows all the columns.
 - Click the Narrow State button to reduce the width of the Library panel.

Change the width of columns

- ❖ Position the pointer between column headers and drag to resize.

You cannot change the order of columns.

Access the Panel menu for the Library panel

- ❖ Click the Panel menu button in the Library panel's title bar.

Work with library items

When you select an item in the Library panel, a thumbnail preview of the item appears at the top of the Library panel. If the selected item is animated or is a sound file, you can use the Play button in the library preview window or the Controller to preview the item.

Use a library item in the current document

- ❖ Drag the item from the Library panel onto the Stage.

The item is added to the current layer.

Convert an object on the Stage to a symbol in the library

- ❖ Drag the item from the Stage onto the current Library panel.

Use a library item from the current document in another document

- ❖ Drag the item from the Library panel or Stage into the Library panel or Stage of another document.

Copy library items from a different document

- 1 Select the document that contains the library items.
- 2 Select the library items in the Library panel.
- 3 Select Edit > Copy.
- 4 Select the document that you want to copy the library items to.
- 5 Select that document's Library panel.
- 6 Select Edit > Paste.

Work with folders in the Library panel

You can organize items in the Library panel using folders. When you create a new symbol, it is stored in the selected folder. If no folder is selected, the symbol is stored at the root of the library.

Create a new folder

- ❖ Click the New Folder button  at the bottom of the Library panel.

Open or close a folder

- ❖ Double-click the folder, or Select the folder and select Expand Folder or Collapse Folder from the Panel menu for the Library panel.

Open or close all folders

- ❖ Select Expand All Folders or Collapse All Folders from the Panel menu for the Library panel.

Move an item between folders

- ❖ Drag the item from one folder to another.

If an item with the same name exists in the new location, Flash prompts you to replace it with the item you are moving.

Sort items in the Library panel

Columns in the Library panel list the name of an item, its type, the number of times it's used in the file, its linkage status and identifier (if the item is associated with a shared library or is exported for ActionScript), and the date on which it was last modified.

You can sort items in the Library panel alphanumerically by any column. Items are sorted within folders.

- ❖ Click the column header to sort by that column. Click the triangle button to the right of the column headers to reverse the sort order.

Edit a library item

- 1 Select the item in the Library panel.
- 2 Select one of the following from the Panel menu for the Library panel:
 - To edit an item in Flash, select Edit.
 - To edit an item in another application, select Edit With and then select an external application.

Note: When starting a supported external editor, Flash opens the original imported document.

Rename a library item

Changing the library item name of an imported file does not change the filename.

- 1 Do one of the following:
 - Double-click the item's name.
 - Select the item and select Rename from the Panel menu for the Library panel.
 - Right-click (Windows) or Control-click (Macintosh) the item and select Rename from the context menu.
- 2 Enter the new name in the box.

Delete a library item

When you delete an item from the library, all instances or occurrences of that item in the document are also deleted.

- ❖ Select the item and click the Trash Can icon at the bottom of the Library panel.

Find unused library items

To organize your document, you can find unused library items and delete them.

Note: It is not necessary to delete unused library items to reduce a Flash document's file size, because unused library items are not included in the SWF file. However, items linked for export are included in the SWF file.

- ❖ Do one of the following:
 - Select Unused Items from the Panel menu for the Library panel.
 - Sort library items by the Use Count column, which indicates whether an item is in use.

See also

[“Sharing library assets”](#) on page 162

Update imported files in the library

If you use an external editor to modify files that you have imported into Flash, such as bitmaps or sound files, you can update the files in Flash without reimporting them. You can also update symbols that you have imported from external Flash documents. Updating an imported file replaces its contents with the contents of the external file.

- 1 Select the imported file in the Library panel.
- 2 Select Update from the Panel menu for the Library panel.

Work with common libraries

You can use the sample common libraries included with Flash to add buttons or sounds to your documents. You can also create custom common libraries, which you can then use with any documents that you create.

See also

[“Configuration folders installed with Flash”](#) on page 377

Use an item from a common library in a document

- 1 Select Window > Common Libraries, and select a library from the submenu.
- 2 Drag an item from the common library into the library for the current document.

Create a common library for your SWF application

- 1 Create a Flash file with a library containing the symbols that you want to include in the common library.
- 2 Place the Flash file in the user-level Libraries folder on your hard disk.
 - On Windows® XP, the path is C:\Documents and Settings*username*\Local Settings\Application Data\Adobe\Flash CS4*language*\Configuration\Libraries\.
 - On Windows® Vista®, the path is C:\Users*username*\Local Settings\Application Data\Adobe\Flash CS4*language*\Configuration\Libraries\.
 - On Mac OS, the path is Hard Disk/Users/*username*/Library/Application Support/Adobe/Flash CS4/*language*/Configuration/Libraries/.

Copy library assets between documents

You can copy library assets from a source document into a destination document in a variety of ways. You can also share symbols between documents as shared library assets during authoring or at runtime.

If you attempt to copy assets that have the same name as existing assets in the destination document, the Resolve Library Conflicts dialog box lets you choose whether to overwrite the existing assets or to preserve the existing assets and add the new assets with modified names. Organize library assets in folders to minimize name conflicts when copying assets between documents.

See also

“[Work with folders in the Library panel](#)” on page 158

“[Working with button symbols](#)” on page 164

“[Sharing library assets](#)” on page 162

Copy a library asset by copying and pasting

- 1 Select the asset on the Stage in the source document.
- 2 Select Edit > Copy.
- 3 Make the destination document the active document.
- 4 To paste the asset in the center of the visible pasteboard, place the pointer on the Stage and select Edit > Paste In Center. To place the asset in the same location as in the source document, select Edit > Paste In Place.

Copy a library asset by dragging

- ❖ With the destination document open, select the asset in the Library panel in the source document and drag the asset into the Library panel in the destination document.

Copy a library asset by opening the source document library in the destination document

- 1 With the destination document active, select File > Import > Open External Library.
- 2 Select the source document, and click Open.
- 3 Drag an asset from the source document library onto the Stage or into the library of the destination document.

Conflicts between library assets

If you import or copy a library asset into a document that already contains a different asset of the same name, choose whether to replace the existing item with the new item. This option is available with all the methods for importing or copying library assets.

The Resolve Library Items dialog box appears when you attempt to place items that conflict with existing items in a document. A conflict exists when you copy an item from a source document that already exists in the destination document and the items have different modification dates. Avoid naming conflicts by organizing your assets inside folders in your document’s library. The dialog box also appears when you paste a symbol or component into your document’s Stage and you already have a copy of the symbol or component that has a different modification date from the one you’re pasting.

If you choose not to replace the existing items, Flash attempts to use the existing item instead of the conflicting item that you are pasting. For example, if you copy a symbol named Symbol 1 and paste the copy into the Stage of a document that already contains a symbol named Symbol 1, Flash creates an instance of the existing Symbol 1.

If you choose to replace the existing items, Flash replaces the existing items (and all their instances) with the new items of the same name. If you cancel the Import or Copy operation, the operation is canceled for all items (not just those items that conflict in the destination document).

Only identical library item types may be replaced with each other. That is, you cannot replace a sound named Test with a bitmap named Test. In such cases, the new items are added to the library with the word Copy appended to the name.

Note: Replacing library items using this method is not reversible. Save a backup of your FLA file before you perform complex paste operations that are resolved by replacing conflicting library items.

If the Resolve Library Conflict dialog box appears when you are importing or copying library assets into a document, resolve the naming conflict.

Resolve naming conflicts between library assets

- ❖ Do one of the following in the Resolve Library Conflict dialog box:
 - To preserve the existing assets in the destination document, click Don't Replace Existing Items.
 - To replace the existing assets and their instances with the new items of the same name, click Replace Existing Items.

Sharing library assets

About shared library assets

Shared library assets let you use assets from a source document in multiple destination documents:

- For runtime shared assets, assets from a source document are linked as external files in a destination document. Runtime assets are loaded into the destination document during document playback—that is, at runtime. The source document containing the shared asset does not need to be available on your local network when you author the destination document. The source document must be posted to a URL for the shared asset to be available to the destination document at runtime.
- For shared assets during authoring, update or replace any symbol in a document you are authoring with any other symbol available on your local network. Update the symbol in the destination document as you author the document. The symbol in the destination document retains its original name and properties, but its contents are updated or replaced with those of the symbol you select.

Using shared library assets can optimize workflow and document asset management.

Working with runtime shared assets

Using runtime shared library assets involves two procedures. First, the author of the source document defines a shared asset in the source document and enters an identifier string for the asset and a URL (*HTTP or HTTPS only*) where the source document will be posted.

Second, the author of the destination document defines a shared asset in the destination document and enters an identifier string and URL identical to those used for the shared asset in the source document. Alternatively, the destination document author can drag the shared assets from the posted source document into the destination document library. The ActionScript version set in the Publish settings must match that of the source document.

In either scenario, the source document must be posted to the specified URL for the shared assets to be available for the destination document.

Define runtime shared assets in a source document

To define sharing properties for an asset in a source document and make the asset accessible for linking to destination documents, use the Symbol Properties dialog box or the Linkage Properties dialog box.

- 1 With the source document open, select Window > Library:
- 2 Do one of the following:
 - Select a movie clip, button, or graphic symbol in the Library panel, and select Properties from the Library Panel menu. Click Advanced.

- Select a font symbol, sound, or bitmap, and select Linkage from the Library Panel menu.
- 3 For Linkage, select Export For Runtime Sharing to make the asset available for linking to the destination document.
 - 4 Enter an identifier for the symbol. Do not include spaces. This is the name Flash uses to identify the asset when linking to the destination document.

Note: Flash also uses the linkage identifier to identify a movie clip or button that is used as an object in ActionScript. See *Working with movie clips in Learning ActionScript 2.0 in Adobe Flash* or *Working with movie clips in Programming ActionScript 3.0*.

- 5 Enter the URL where the SWF file containing the shared asset will be posted, and click OK.

When you publish the SWF file, you must post the SWF file to the URL you specified so that the shared assets are available to destination documents.

Link to runtime shared assets from a destination document

You can link to a shared asset by entering its URL or by dragging the asset into the destination document.

Link a shared asset to a destination document by entering the identifier and URL

- 1 In the destination document, select Window > Library.
- 2 Do one of the following:
 - Select a movie clip, button, graphic symbol, bitmap, or sound in the Library panel, and select Properties from the Library Panel menu. Click Advanced.
 - Select a font symbol, and select Linkage from the Library Panel menu.
- 3 For Linkage, select Import For Runtime Sharing to link to the asset in the source document.
- 4 Enter an identifier for the symbol, bitmap, or sound that is identical to the identifier used for the symbol in the source document. Do not include spaces.
- 5 Enter the URL where the SWF source file containing the shared asset is posted, and click OK.

Link a shared asset to a destination document by dragging

- 1 In the destination document, do one of the following:
 - Select File > Open.
 - Select File > Import > Open External Library.
- 2 Select the source document and click Open.
- 3 Drag the shared asset from the source document Library panel into the Library panel or onto the Stage in the destination document.

Turn off sharing for a symbol in a destination document

- 1 In the destination document, select the linked symbol in the Library panel and do one of the following:
 - If the asset is a movie clip, button, or graphic symbol, select Properties from the Library Panel menu.
 - If the asset is a font symbol, select Linkage from the Library Panel menu.
- 2 Deselect Import For Runtime Sharing, and click OK.

Update or replace symbols

You can update or replace a movie clip, button, or graphic symbol in a document with any other symbol in a FLA file accessible on your local network. The original name and properties of the symbol in the destination document are preserved, but the contents of the symbol are replaced with the contents of the symbol you select. Any assets that the selected symbol uses are also copied into the destination document.

- 1 With the document open, select a movie clip, button, or graphic symbol and select Properties from the Library Panel menu.
- 2 If the Linkage and Source areas of the Symbol Properties dialog box are not showing, click Advanced.
- 3 To select a new FLA file, click Browse.
- 4 Navigate to a FLA file that contains the symbol to use to update or replace the selected symbol in the Library panel, and click Open.
- 5 Navigate to a symbol, and click OK.
- 6 In the Symbol Properties dialog box, under Source, select Always Update Before Publishing and click OK.

Working with button symbols

Create a button

Buttons are actually four-frame interactive movie clips. When you select the button behavior for a symbol, Flash creates a Timeline with four frames. The first three frames display the button's three possible states; the fourth frame defines the active area of the button. The Timeline doesn't actually play; it reacts to pointer movement and actions by jumping to the appropriate frame.

To make a button interactive, you place an instance of the button symbol on the Stage and assign actions to the instance. You must assign the actions to the instance of the button in the document, not to frames in the button's Timeline.

Each frame in the Timeline of a button symbol has a specific function:

- The first frame is the Up state, representing the button whenever the pointer is not over the button.
- The second frame is the Over state, representing the button's appearance when the pointer is over the button.
- The third frame is the Down state, representing the button's appearance as it is clicked.
- The fourth frame is the Hit state, defining the area that responds to the mouse click. This area is invisible in the SWF file.

You create a button using a movie clip symbol or a button component. Using each type of button has advantages. Creating a button using a movie clip lets you add more frames to the button or add more complex animation. However, movie clip buttons have a larger file size than button symbols. Using a button component allows you to bind the button to other components and share and display data in an application. Button components also include prebuilt features, such as accessibility support, and can be customized. Button components include the Button, RadioButton, and CheckBox. For more information, see Button component in *ActionScript 2.0 Components Language Reference* at http://www.adobe.com/go/learn_fl_cs4_as2clr_en or Use the Button component in *Using ActionScript 3.0 Components*.

To create a button:

- 1 Select Edit > Deselect All to ensure that nothing is selected on the Stage.

- 2 Select Insert > New Symbol, or press Control+F8 (Windows) or Command+F8 (Macintosh).

To create the button, you convert the button frames to keyframes.

- 3 In the Create New Symbol dialog box, enter a name for the new button symbol. For the symbol Type, select Button. Flash switches to symbol-editing mode. The Timeline header changes to display four consecutive frames labeled Up, Over, Down, and Hit. The first frame, Up, is a blank keyframe.

- 4 To create the Up state button image, select the Up frame in the Timeline and then use the drawing tools, import a graphic, or place an instance of another symbol on the Stage.

You can use a graphic or movie clip symbol in a button, but you cannot use another button in a button. Use a movie clip symbol to animate the button.

- 5 Click the Over frame, and select Insert > Timeline > Keyframe.

Flash inserts a keyframe that duplicates the contents of the Up frame.

- 6 Change or edit the button image for the Over state.

- 7 Repeat steps 5 and 6 for the Down frame and the Hit frames.

The Hit frame is not visible on the Stage, but it defines the area of the button that responds when clicked. The graphic for the Hit frame must be a solid area large enough to encompass all the graphic elements of the Up, Down, and Over frames. It can also be larger than the visible button. If you do not specify a Hit frame, the image for the Up state is used as the Hit frame.

To create a disjoint rollover, in which moving the pointer over a button causes another graphic on the Stage to change, place the Hit frame in a different location than the other button frames.

- 8 To assign a sound to a state of the button, select that state's frame in the Timeline, select Window > Properties, and then select a sound from the Sound menu in the Property inspector.
- 9 When you finish, select Edit > Edit Document. To create an instance of the button in the document, drag the button symbol from the Library panel.

See also

[“Using sounds in Flash”](#) on page 273

Enable, edit, and test buttons

By default, Flash keeps buttons disabled as you create them, to make it easier to select and work with them. When a button is disabled, clicking the button selects it. When a button is enabled, it responds to the mouse events that you've specified as if the SWF file were playing. You can still select enabled buttons. Disable buttons as you work, and enable buttons to quickly test their behavior.

Enable and disable buttons

- ❖ Select Control > Enable Simple Buttons. A check mark appears next to the command to indicate buttons are enabled. Select the command again to disable buttons.

Any buttons on the Stage now respond. As you move the pointer over a button, Flash displays the Over frame; when you click within the button's active area, Flash displays the Down frame.

Select, move, or edit an enabled button

- ❖ Do one of the following:
 - Use the Selection tool to drag a selection rectangle around the button.

- Use the arrow keys to move the button.
- If the Property inspector is not visible, select Window > Properties to edit the button in the Property inspector, or Alt+double-click (Windows) or Option+double-click (Macintosh) the button.

Test a button

- ❖ Do one of the following:
 - Select Control > Enable Simple Buttons. Move the pointer over the enabled button to test it.
 - Select the button in the Library panel, and click the Play button in the Library preview window.
 - Select Control > Test Scene or Control > Test Movie.

Movie clips in buttons are not visible in the Flash authoring environment.

Scaling and caching symbols

About 9-slice scaling and movie clip symbols

9-slice scaling allows you to specify how scaling is applied to specific areas of a movie clip. With 9-slice scaling, you can ensure that the movie clip looks correct when scaled. With normal scaling, Flash scales all parts of a movie clip equally, and in both the horizontal and vertical dimensions. For many movie clips, this equal scaling can make the clip's graphics look strange, especially at the corners of rectangular movie clips. This is often true of movie clips used as user interface elements, such as buttons.

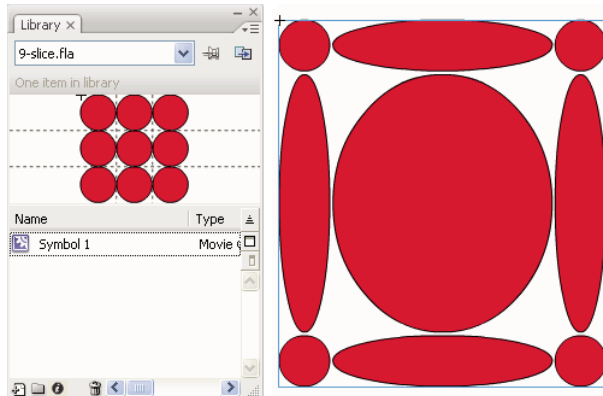
The movie clip is visually divided into nine sections with a grid-like overlay, and each of the nine areas is scaled independently. To maintain the visual integrity of the movie clip, corners are not scaled, while the remaining areas of the image are scaled (as opposed to being stretched) larger or smaller, as needed.

When a movie clip symbol has 9-slice scaling applied, it appears in the Library panel preview with the guides displayed. If Enable Live Preview is turned on (Control > Enable Live Preview) when you scale instances of the movie clip on the Stage, you see the 9-slice scaling applied on the Stage.

Note: 9-slice scaling cannot be applied to Graphic or Button symbols. Bitmaps inside 9-slice enabled movie clips are scaled normally, without 9-slice distortion, while the other movie clip contents are scaled according to the 9-slice guides.

Note: 9-slice scaling is sometimes also referred to as "scale 9."

A 9-slice-enabled movie clip can contain nested objects within it, but only certain types of objects inside the movie clip properly scale in the 9-slice manner. To make a movie clip with internal objects that also adhere to 9-slice scaling when the movie clip is scaled, those nested objects must be shapes, drawing objects, groups, or graphic symbols.



A 9-slice-enabled symbol in the Library panel and scaled on the Stage

For video tutorials about 9-slice scaling, see:

- www.adobe.com/go/vid0204
- www.adobe.com/go/vid0205

Edit movie clip symbols with 9-slice scaling

By default, slice guides are placed at 25% (or one-fourth) of the symbol's width and height from the edge of the symbol. In symbol-editing mode, the slice guides appear as dotted lines superimposed on the symbol. The slice guides don't snap when you drag them on the pasteboard. The guides do not appear when the symbol is on the Stage.

You cannot edit 9-slice-enabled symbols in place on the Stage. You must edit them in symbol-editing mode.

Note: Instances made from a 9-slice-enabled movie clip symbol can be transformed, but should not be edited. Editing these instances can have unpredictable results.

For video tutorials about 9-slice scaling, see:

- www.adobe.com/go/vid0204
- www.adobe.com/go/vid0205

Enable 9-slice scaling for an existing movie clip symbol

- 1 With the source document open, select Window > Library.
- 2 Select a movie clip, button, or graphic symbol in the Library panel.
- 3 Select Properties from the Library Panel menu.
- 4 Select Enable Guides for 9-slice Scaling.

Edit a 9-slice-enabled movie clip symbol

- 1 Enter symbol-editing mode by doing one of the following:
 - Select an instance of the symbol on the Stage and right-click (Windows) or Control-click (Macintosh), and select Edit.
 - Select the symbol in the Library panel and right-click (Windows) or Control-click (Macintosh), and select Edit.
 - Double-click the symbol in the Library panel.
- 2 To move the horizontal or vertical guides, drag and release a guide. The new position of the guide is updated in the Library preview for the symbol.

Using runtime bitmap caching with movie clip and button symbols

Runtime bitmap caching lets you optimize playback performance by specifying that a static movie clip (for example, a background image) or button symbol be cached as a bitmap at runtime. Caching a movie clip as a bitmap prevents Flash Player from having to continually redraw the image, which provides a significant improvement in playback performance.

For example, when you create animations with a complex background, create a movie clip for the background. The background is rendered as a bitmap stored at the current screen depth. It can be drawn quickly, letting the animation play faster and more smoothly.

Without bitmap caching, the animation might play back too slowly.

Bitmap caching lets you use a movie clip and freeze it in place automatically. If a region changes, vector data updates the bitmap cache. This process minimizes the number of redraws that Flash Player must perform, and provides smoother, faster playback performance.

Only use runtime bitmap caching on static, complex movie clips in which the position, but not the content, of the movie clip changes on each frame in an animation. The playback or runtime performance improvement from using runtime bitmap caching is only noticeable on complex-content movie clips. Runtime bitmap caching with simple movie clips does not enhance performance.

For more information, see [When to enable caching in *Learning ActionScript 2.0 in Adobe Flash*](#).

Note: *You can only use the Use Runtime Bitmap Caching option for movie clip and button symbols.*

Under the following circumstances, a movie clip does not use a bitmap (even if Use Runtime Bitmap Caching is selected) but instead renders the movie clip or button symbol by using vector data:

- The bitmap is too large (greater than 2880 pixels in either direction).
- The bitmap fails to allocate (producing an out-of-memory error).

Specify bitmap caching for a movie clip

- 1 Select the movie clip or button symbol on the Stage.
- 2 In the Property inspector, select Use Runtime Bitmap Caching.

Symbols and ActionScript

With ActionScript®, you can control symbols at runtime. Using ActionScript allows you to create interaction and other capabilities in your FLA files that are not possible with the Timeline alone.

Controlling instances and symbols with ActionScript

To control movie clip and button instances, use ActionScript®. The movie clip or button instance must have a unique instance name to be used with ActionScript. You can write the ActionScript yourself or use the pre-defined behaviors included with Flash.

For more information, see Handling events in [Learning ActionScript 2.0 in Adobe Flash](#) or Handling events in [Programming ActionScript 3.0](#).

See also

“[Edit symbols](#)” on page 150

Controlling instances with behaviors

In FLA files where the ActionScript Publish setting is set to ActionScript 2.0, you can use behaviors to control movie clip and graphic instances in a document without writing ActionScript. Behaviors are prewritten ActionScript scripts that let you add ActionScript coding to your document without having to create the ActionScript code yourself.

Behaviors are not available for ActionScript 3.0.

You can use behaviors with an instance to arrange it in the stacking order on a frame, as well as to load or unload, play, stop, duplicate, or drag a movie clip, or to link to a URL.

In addition, you can use behaviors to load an external graphic or an animated mask into a movie clip.

Flash includes the behaviors in the following table.

Behavior	Purpose	Select or input
Load Graphic	Loads an external JPEG file into a movie clip or screen.	Path and filename of JPEG file. Instance name of movie clip or screen receiving the graphic.
Load External Movieclip	Loads an external SWF file into a target movie clip or screen.	URL of external SWF file. Instance name of movie clip or screen receiving the SWF file.
Duplicate Movieclip	Duplicates a movie clip or screen.	Instance name of movie clip to duplicate. X-offset and Y-offset of pixels from original to copy.
Goto And Play at frame or label	Plays a movie clip from a particular frame.	Instance name of target clip to play. Frame number or label to play.
Goto And Stop at frame or label	Stops a movie clip, optionally moving the playhead to a particular frame.	Instance name of target clip to stop. Frame number or label to stop.
Bring To Front	Brings target movie clip or screen to the top of the stacking order.	Instance name of movie clip or screen.
Bring Forward	Brings target movie clip or screen one position higher in the stacking order.	Instance name of movie clip or screen.
Send To Back	Sends the target movie clip to the bottom of the stacking order.	Instance name of movie clip or screen.
Send Backward	Sends the target movie clip or screen one position lower in the stacking order.	Instance name of movie clip or screen.
Start Dragging Movieclip	Starts dragging a movie clip.	Instance name of movie clip or screen.
Stop Dragging Movieclip	Stops the current drag.	
Unload Movieclip	Removes a movie clip that was loaded by means of loadMovie() from Flash Player.	Instance name of movie clip.

See also

[“Control sounds using behaviors”](#) on page 280

[“Control video playback using behaviors”](#) on page 298

Add and configure a behavior

Be sure you are working in a FLA file whose ActionScript Publish setting is ActionScript 2.0 or earlier.

- 1 Select the object, such as a button, to trigger the behavior.
- 2 In the Behaviors panel (Window > Behaviors), click the Add (+) button and select the desired behavior from the Movieclip submenu.
- 3 Select the movie clip to control with the behavior.
- 4 Select a relative or absolute path.
- 5 If required, select or input settings for the behavior parameters and click OK. Default settings for the behavior appear in the Behaviors panel.
- 6 Under Event, click On Release (the default event) and select a mouse event from the menu. To use the On Release event, leave the option unchanged.

See also

[“Relative paths”](#) on page 180

[“Absolute paths”](#) on page 180

Create custom behaviors

To write custom behaviors, create an XML file that contains the ActionScript 2.0 code to perform the desired behavior, and save the file in the Behaviors folder of your local computer. Behaviors are stored in the following location:

- Windows XP: C:\Documents and Settings*user name*\Local Settings\Application Data\Adobe\Flash CS3*language*\Configuration\Behaviors
- Windows Vista: C:\Users*user name*\Local Settings\Application Data\Adobe\Flash CS3*language*\Configuration\Behaviors
- Macintosh: Macintosh HD/Users/*user name*/Library/Application Support/Adobe/Flash CS3/*language*/Configuration/Behaviors/

Before you create your own behaviors, examine the Behavior XML files to develop an understanding of the syntax of the XML files, as well as the ActionScript code used to create behaviors. If you are new to writing behaviors, familiarize yourself with the XML tags used to create user interface elements (such as dialog boxes), and with ActionScript, the coding language used to create behaviors. To learn about the XML used to create interface elements, see *Extending Flash*. To learn about ActionScript, see *Programming ActionScript 3.0* or *Learning ActionScript 2.0 in Adobe Flash*.

You can also download behaviors that other Flash users have created from the Adobe Flash Exchange website. You can visit the Adobe Exchange at: www.adobe.com/go/flash_exchange.

- 1 Using an XML editor, open an existing behavior’s XML file, and rename the file appropriately for the behavior you intend to create.
- 2 Enter a new value for the `category` attribute of the `behavior_definition` tag in the XML file.

The following XML code creates a category named myCategory in the Flash Behaviors panel under which the behavior will be listed.

```
<behavior_definition dialogID="Trigger-dialog" category="myCategory"
authoringEdition="pro" name="behaviorName">
```

- 3 Enter a new value for the name attribute of the behavior_definition tag. This will be the name of the behavior as it will appear in the Flash authoring environment.
- 4 (Optional) If your custom behavior requires a dialog box, enter parameters using the <properties> and <dialog> tags.

To learn about the tags and parameters used to create your own custom dialog boxes, see *Extending Flash*.

- 5 In the <actionscript> tag, insert the ActionScript code to create the behavior.

If you are new to ActionScript, see *Learning ActionScript 2.0 in Adobe Flash* or *Programming ActionScript 3.0*.

For example (from the Movieclip_loadMovie.xml behavior file) (ActionScript 2.0):

```
<actionscript>
  <![CDATA[ //load Movie Behavior
    if($target$ == Number($target$)){
      loadMovieNum($clip$, $target$);
    } else {
      $target$.loadMovie($clip$);
    }
  //End Behavior
  ]]>
</actionscript>
```

- 6 Save the file and test the behavior.

See also

[“Break apart an instance symbol”](#) on page 155

Chapter 7: Timelines and Animation

Working with timelines

Frames and keyframes

Like films, Adobe® Flash® CS4 Professional documents divide lengths of time into frames. In the Timeline, you work with these frames to organize and control the content of your document. You place frames in the Timeline in the order you want the objects in the frames to appear in your finished content.

A *keyframe* is a frame where a new symbol instance appears in the Timeline. A keyframe can also be a frame that includes ActionScript® code to control some aspect of your document. You can also add a *blank keyframe* to the Timeline as a placeholder for symbols you plan to add later or to explicitly leave the frame blank.

A *property keyframe* is a frame in which you define a change to an object's properties for an animation. Flash can *tween*, or automatically fill in, the property values between the property keyframes in order to produce fluid animations. Because property keyframes let you produce animation without drawing each individual frame, they make creating animation easier. A series of frames containing tweened animation is called a *motion tween*.

A *tweened frame* is any frame that is part of a motion tween.

A *static frame* is any frame that is not part of a motion tween.

You arrange keyframes and property keyframes in the Timeline to control the sequence of events in your document and its animation.

See also

[“The Timeline”](#) on page 22

[“Animation basics”](#) on page 183

[“Motion tweens”](#) on page 187

Insert frames in the Timeline

- To insert a new frame, select Insert > Timeline > Frame.
- To create a new keyframe, select Insert > Timeline > Keyframe, or right-click (Windows) or Control-click (Macintosh) the frame where you want to place a keyframe, and select Insert Keyframe from the context menu.
- To create a new blank keyframe, select Insert > Timeline > Blank Keyframe, or right-click (Windows) or Control-click (Macintosh) the frame where you want to place the keyframe, and select Insert Blank Keyframe from the context menu.

Select frames in the Timeline

Flash offers two different methods for selecting frames in the Timeline. In frame-based selection (the default), you select individual frames in the Timeline. In span-based selection, the entire frame sequence, from one keyframe to the next, is selected when you click any frame in the sequence. You can specify span-based selection in Flash Preferences.

- To select one frame, click the frame. If you have Span Based Selection enabled, Control-click (Windows) or Command-click (Macintosh) the frame.
- To select multiple contiguous frames, Shift-click additional frames.

- To select multiple non-contiguous frames, Control-click (Windows) or Command-click (Macintosh) additional frames.
- To select all frames in the Timeline, select Edit > Timeline > Select All Frames.
- To select an entire span of static frames, double-click a frame between two keyframes. If you have Span Based Selection enabled, click any frame in the sequence.

Specify span-based frame selection

- 1 Select Edit > Preferences.
- 2 Select the General category.
- 3 In the Timeline section, select Span Based Selection.
- 4 Click OK.

Copy or paste a frame or frame sequence

- ❖ Do one of the following:
 - Select the frame or sequence and select Edit > Timeline > Copy Frames. Select the frame or sequence that you want to replace, and select Edit > Timeline > Paste Frames.
 - Alt-click (Windows) or Option-click (Macintosh) and drag the keyframe to the location where you want to paste it.

Delete a frame or frame sequence

- ❖ Select the frame or sequence and select Edit > Timeline > Remove Frame, or right-click (Windows) or Control-click (Macintosh) the frame or sequence and select Remove Frame from the context menu.

Surrounding frames remain unchanged.

Move a keyframe or frame sequence and its contents

- ❖ Drag the keyframe or sequence to the desired location.

Change the length of a static frame sequence

- ❖ Control-drag (Windows) or Command-drag (Macintosh) the beginning or ending frame of the span to the left or right.

To change the length of a frame-by-frame animation sequence, see “[Create frame-by-frame animations](#)” on page 220.

Convert a keyframe to a frame

- ❖ Select the keyframe and select Edit > Timeline > Clear Keyframe, or right-click (Windows) or Control-click (Macintosh) the keyframe and select Clear Keyframe from the context menu.

The Stage contents of the cleared keyframe and all frames up to the subsequent keyframe are replaced with the Stage contents of the frame preceding the cleared keyframe.

Create and organize layers

Layers help you organize the artwork in your document. You can draw and edit objects on one layer without affecting objects on another layer. In areas of the Stage with nothing on a layer, you can see through it to the layers below.

To draw, paint, or otherwise modify a layer or folder, select the layer in the Timeline to make it active. A pencil icon next to a layer or folder name in the Timeline indicates that the layer or folder is active. Only one layer can be active at a time (although more than one layer can be selected at a time).

When you create a Flash document, it contains only one layer. To organize the artwork, animation, and other elements in your document, add more layers. You can also hide, lock, or rearrange layers. The number of layers you can create is limited only by your computer's memory, and layers do not increase the file size of your published SWF file. Only the objects you place into layers add to the file size.

To organize and manage layers, create layer folders and place layers in them. You can expand or collapse layer folders in the Timeline without affecting what you see on the Stage. Use separate layers or folders for sound files, ActionScript, frame labels, and frame comments. This helps you find these items quickly to edit them.

To help create sophisticated effects, use special guide layers to make drawing and editing easier, and to make mask layers.

There are five types of layers you can use in Flash:

- Normal layers contain most of the artwork in a FLA file.
- Mask layers contain objects used as masks to hide selected portions of layers below them. For more information, see [“Using mask layers”](#) on page 221.
- Masked layers are layers beneath a mask layer that you associate with the mask layer. Only the portion of the mask layer not covered by the mask is visible. For more information, see [“Using mask layers”](#) on page 221.
- Guide layers contain strokes that can be used to guide the arrangement of objects on other layers or the motion of classic tween animations on other layers. For more information, see [“Guide layers”](#) on page 177 and [“Create classic tween motion along a path”](#) on page 213.
- Guided layers are layers associated with a guide layer. The objects on the guided layer can be arranged or animated along the strokes on the guide layer. Guided layers can contain static artwork and classic tweens, but not motion tweens.

Normal, Mask, Masked, and Guide layers can contain motion tweens or inverse kinematic bones. When these items are present in one of these layers, there are limitations to the types of content that can be added to the layer. For more information, see [“Motion tweens”](#) on page 187 and [“Using inverse kinematics”](#) on page 223.


See also

[“Change layer height in the Timeline”](#) on page 24

Create a layer

When you create a layer, it appears above the selected layer. The newly added layer becomes the active layer.

❖ Do one of the following:

- Click the New Layer button  at the bottom of the Timeline.
- Select Insert > Timeline > Layer.
- Right-click (Windows) or Control-click (Macintosh) a layer name in the Timeline and select Insert Layer from the context menu.

Create a layer folder

❖ Do one of the following:

- Select a layer or folder in the Timeline and select Insert > Timeline > Layer Folder.
- Right-click (Windows) or Control-click (Macintosh) a layer name in the Timeline and select Insert Folder from the context menu. The new folder appears above the layer or folder you selected.

- Click the New Folder icon  at the bottom of the Timeline. The new folder appears above the layer or folder you selected.

Organize layers and layer folders

To organize your document, rearrange layers and folders in the Timeline.

Layer folders help organize your workflow by letting you place layers in a tree structure. To see the layers a folder contains without affecting which layers are visible on the Stage, expand or collapse the folder. Folders can contain both layers and other folders, allowing you to organize layers in much the same way you organize files on your computer.

The layer controls in the Timeline affect all layers within a folder. For example, locking a layer folder locks all layers within that folder.

- To move a layer or layer folder into a layer folder, drag the layer or layer folder name to the destination layer folder name.
- To change the order of layers or folders, drag one or more layers or folders in the Timeline to the desired position.
- To expand or collapse a folder, click the triangle to the left of the folder name.
- To expand or collapse all folders, Right-click (Windows) or Control-click (Macintosh) and select Expand All Folders or Collapse All Folders.

Rename a layer or folder

By default, new layers are named by the order in which they are created: Layer 1, Layer 2, and so on. To better reflect their contents, rename layers.

- ❖ Do one of the following:
 - Double-click the name of the layer or folder in the Timeline and enter a new name.
 - Right-click (Windows) or Control-click (Macintosh) the name of the layer or folder and select Properties from the context menu. Enter the new name in the Name box and click OK.
 - Select the layer or folder in the Timeline and select Modify > Timeline > Layer Properties. Enter the new name in the Name box and click OK.

Select a layer or folder

- ❖ Do one of the following:
 - Click the name of a layer or folder in the Timeline.
 - Click any frame in the Timeline of the layer to select.
 - Select an object on the Stage that is located in the layer to select.
 - To select contiguous layers or folders, Shift-click their names in the Timeline.
 - To select non-contiguous layers or folders, Control-click (Windows) or Command-click (Macintosh) their names in the Timeline.

Copy a layer

- 1 To select the entire layer, click the layer name in the Timeline.
- 2 To create a layer, click the Insert Layer button.
- 3 Select Edit > Timeline > Copy Frames.
- 4 Click the new layer and select Edit > Timeline > Paste Frames.

Copy the contents of a layer folder

- 1 Collapse the folder (click the triangle to the left of the folder name in the Timeline) and click the folder name to select the entire folder.
- 2 Select Edit > Timeline > Copy Frames.
- 3 To create a folder, select Insert > Timeline > Layer Folder.
- 4 Click the new folder and select Edit > Timeline > Paste Frames.

Delete a layer or folder

- 1 To select the layer or folder, click its name in the Timeline or any frame in the layer.
- 2 Do one of the following:
 - Click the Delete Layer button in the Timeline.
 - Drag the layer or folder to the Delete Layer button.
 - Right-click (Windows) or Control-click (Macintosh) the layer or folder name and select Delete Layer from the context menu.

Note: When you delete a layer folder, all the enclosed layers and all their contents are also deleted.

Lock or unlock one or more layers or folders

- To lock a layer or folder, click in the Lock column to the right of the name. To unlock the layer or folder, click in the Lock column again.
- To lock all layers and folders, click the padlock icon. To unlock all layers and folders, click it again.
- To lock or unlock multiple layers or folders, drag through the Lock column.
- To lock all *other* layers or folders, Alt-click (Windows) or Option-click (Macintosh) in the Lock column to the right of a layer or folder name. To unlock all layers or folders, Alt-click or Option-click in the Lock column again.

View layers and layer folders

Show or hide a layer or folder

A red X next to the name of a layer or folder in the Timeline indicates that a layer or folder is hidden. In the publish settings, you can choose whether hidden layers are included when you publish a SWF file.

- To hide a layer or folder, click in the Eye column to the right of the layer or folder name in the Timeline. To show the layer or folder, click in it again.
- To hide all the layers and folders in the Timeline, click the Eye icon. To show all layers and folders, click it again.
- To show or hide multiple layers or folders, drag through the Eye column.
- To hide all layers and folders other than the current layer or folder, Alt-click (Windows) or Option-click (Macintosh) in the Eye column to the right of a layer or folder name. To show all layers and folders, Alt-click or Option-click it again.

View the contents of a layer as outlines

To distinguish which layer an object belongs to, display all objects on a layer as colored outlines.

- To display all objects on that layer as outlines, click in the Outline column to the right of the layer's name. To turn off outline display, click in it again.

- To display objects on all layers as outlines, click the outline icon. To turn off outline display on all layers, click it again.
- To display objects on all layers *other* than the current layer as outlines, Alt-click (Windows) or Option-click (Macintosh) in the Outline column to the right of a layer's name. To turn off the outline display for all layers, Alt-click or Option-click in it again.

Change a layer's outline color

1 Do one of the following:

- Double-click the layer's icon (the icon to the left of the layer name) in the Timeline.
- Right-click (Windows) or Control-click (Macintosh) the layer name and select Properties from the context menu.
- Select the layer in the Timeline and select Modify > Timeline > Layer Properties.

2 In the Layer Properties dialog box, click the Outline Color box, select a new color, and click OK.

Note: Motion paths on the layer also use the layer outline color.

Guide layers

For help in aligning objects when drawing, create guide layers and align objects on other layers to the objects you create on the guide layers. Guide layers are not exported and do not appear in a published SWF file. Any layer can be a guide layer. Guide layers are indicated by a guide icon to the left of the layer name.

See also

[“Align objects”](#) on page 121

[“Motion tweens”](#) on page 187

[“Using inverse kinematics”](#) on page 223

[“Create classic tween motion along a path”](#) on page 213

Create a guide layer

- ❖ Select the layer and Right-click (Windows) or Control-click (Macintosh) and select Guide from the context menu. To change the layer back to a normal layer, select Guide again.

Create a motion guide layer

To control the movement of objects in a *classic tween* animation, create a motion guide layer.

You cannot drag a *motion tween* layer or inverse kinematics *pose layer* onto a guide layer.

- ❖ Drag a normal layer onto a guide layer. This converts the guide layer to a motion guide layer and links the normal layer to the new motion guide layer.

Note: To prevent accidentally converting a guide layer, place all guide layers at the bottom of the layer order.

About multiple timelines and levels

Flash® Player has a stacking order of levels. Every Flash document has a main Timeline located at level 0 in Flash Player. You can use the `loadMovie` command to load other Flash documents (SWF files) into Flash Player at different levels.

If you load documents into levels above level 0, the documents stack on top of one another like drawings on transparent paper; when there is no content on the Stage, you can see through to the content on lower levels. If you load a document into level 0, it replaces the main timeline. Each document loaded into a level of Flash Player has its own timeline.

Timelines can send messages to each other with ActionScript. For example, an action on the last frame of one movie clip can tell another movie clip to play. To use ActionScript to control a timeline, you must use a target path to specify the location of the timeline.

For more information, see the `MovieClip.loadMovie` method in [ActionScript 2.0 Language Reference](#).

About nested movie clips and parent-child hierarchy

When you create a movie clip instance in a Flash document, the movie clip has its own timeline. Every movie clip symbol has its own timeline. The movie clip's timeline is nested inside the main timeline of the document. You can also nest a movie clip instance inside another movie clip symbol.

When a movie clip is created inside a Flash document, or nested inside another movie clip, it becomes a child of that document or movie clip, which becomes the parent. Relationships between nested movie clips are hierarchical: modifications made to the parent affect the child. The root Timeline for each level is the parent of all the movie clips on its level, and because it is the top-most Timeline, it has no parent. In the Movie Explorer panel, you can view the hierarchy of nested movie clips in a document by choosing Show Symbol Definitions from the panel menu.

To understand movie clip hierarchy, consider the hierarchy on a computer: the hard disk has a root directory (or folder) and subdirectories. The root directory is analogous to the main (or root) Timeline of a Flash document: it is the parent of everything else. The subdirectories are analogous to movie clips.

You can use the movie clip hierarchy in Flash to organize related objects. For example, you could create a Flash document containing a car that moves across the Stage. You can use a movie clip symbol to represent the car and set up a motion tween to move it across the Stage.

To add wheels that rotate, you can create a movie clip for a car wheel, and create two instances of this movie clip, named `frontWheel` and `backWheel`. Then you can place the wheels on the car movie clip's Timeline—not on the main Timeline. As children of `car`, `frontWheel` and `backWheel` are affected by any changes made to `car`; they move with the car as it tweens across the Stage.

To make both wheel instances spin, you can set up a motion tween that rotates the wheel symbol. Even after you change `frontWheel` and `backWheel`, they continue to be affected by the tween on their parent movie clip, `car`; the wheels spin, but they also move with the parent movie clip `car` across the Stage.

See also

[“Symbols, instances, and library assets”](#) on page 147

Working with scenes

To organize a document thematically, you can use scenes. For example, you might use separate scenes for an introduction, a loading message, and credits. Though using scenes has some disadvantages, there are some situations in which few of these disadvantages apply, such as when you create lengthy animations. When you use scenes, you avoid having to manage a large number of FLA files because each scene is contained within a single FLA file.

Using scenes is similar to using several FLA files together to create a larger presentation. Each scene has a Timeline. Frames in the document are numbered consecutively through the scenes. For example, if a document contains two scenes with ten frames each, the frames in Scene 2 are numbered 11–20. The scenes in the document play back in the order they are listed in the Scene panel. When the playhead reaches the final frame of a scene, the playhead progresses to the next scene.

Disadvantages of scenes

When you publish a SWF file, the Timeline of each scene combines into a single Timeline in the SWF file. After the SWF file compiles, it behaves as if you created the FLA file using one scene. Because of this behavior, scenes have some disadvantages:

- Scenes can make documents confusing to edit, particularly in multi-author environments. Anyone using the FLA document might have to search several scenes within a FLA file to locate code and assets. Consider loading external SWF content or using movie clips instead.
- Scenes often result in large SWF files. Using scenes encourages you to place more content in a single FLA file, which results in larger FLA files and SWF files.
- Scenes force users to progressively download the entire SWF file, even if they do not plan or want to watch all of it. If you avoid scenes, users can control what content they download as they progress through your SWF file.
- Scenes combined with ActionScript might produce unexpected results. Because each scene Timeline is compressed onto a single Timeline, you might encounter errors involving your ActionScript and scenes, which typically require extra, complicated debugging.

Controlling scene playback

To stop or pause a document after each scene, or to let users navigate the document in a nonlinear fashion, you use ActionScript. For more information see, “[ActionScript](#)” on page 330.


Display the Scene panel

- ❖ Select Window > Other Panels > Scene.

Add a scene

- ❖ Select Insert > Scene, or click the Add Scene button  in the Scene panel.


Delete a scene

- ❖ Click the Delete Scene button  in the Scene panel.

Change the name of a scene

- ❖ Double-click the scene name in the Scene panel and enter the new name.

Duplicate a scene

- ❖ Click the Duplicate Scene button  in the Scene panel.

Change the order of a scene in the document

- ❖ Drag the scene name to a different location in the Scene panel.

View a particular scene

- ❖ Select View > Go To, and then select the name of the scene from the submenu.

Timelines and ActionScript

With ActionScript®, you can control the Timeline at runtime. Using ActionScript allows you to create interaction and other capabilities in your FLA files that is not possible with the Timeline alone.

Absolute paths

An absolute path starts with the name of the level into which the document is loaded and continues through the display list until it reaches the target instance. You can also use the alias `_root` to refer to the topmost Timeline of the current level. For example, an action in the movie clip `california` that refers to the movie clip `oregon` could use the absolute path `_root.westCoast.oregon`.

The first document to open in Flash Player is loaded at level 0. You must assign each additional loaded document a level number. When you use an absolute reference in ActionScript to reference a loaded document, use the form `_levelX`, where `X` is the level number into which the document is loaded. For example, the first document that opens in Flash Player is called `_level0`; a document loaded into level 3 is called `_level3`.

To communicate between documents on different levels, you must use the level name in the target path. The following example shows how the `portland` instance would address the `atlanta` instance located in a movie clip called `georgia` (`georgia` is at the same level as `oregon`):

```
_level5.georgia.atlanta
```

You can use the `_root` alias to refer to the main Timeline of the current level. For the main Timeline, the `_root` alias stands for `_level0` when targeted by a movie clip also on `_level0`. For a document loaded into `_level5`, `_root` is equal to `_level5` when targeted by a movie clip also on level 5. For example, if the movie clips `southcarolina` and `florida` are both loaded into the same level, an action called from the instance `southcarolina` could use the following absolute path to target the instance `florida`:

```
_root.eastCoast.florida
```

Relative paths

A relative path depends on the relationship between the controlling Timeline and the target Timeline. Relative paths can address targets only within their own level of Flash Player. For example, you can't use a relative path in an action on `_level0` that targets a Timeline on `_level5`.

In a relative path, use the keyword `this` to refer to the current Timeline in the current level; use the `_parent` alias to indicate the parent Timeline of the current Timeline. You can use the `_parent` alias repeatedly to go up one level in the movie clip hierarchy within the same level of Flash Player. For example, `_parent._parent` controls a movie clip up two levels in the hierarchy. The topmost Timeline at any level in Flash Player is the only Timeline with a `_parent` value that is undefined.

An action in the Timeline of the instance `charleston`, located one level below `southcarolina`, could use the following target path to target the instance `southcarolina`:

```
_parent
```

To target the instance `eastCoast` (one level up) from an action in `charleston`, you could use the following relative path:

```
_parent._parent
```

To target the instance `atlanta` from an action in the Timeline of `charleston`, you could use the following relative path:

```
_parent._parent.georgia.atlanta
```

Relative paths are useful for reusing scripts. For example, you could attach the following script to a movie clip that magnifies its parent by 150%:

```
onClipEvent (load) {_parent._xscale  
= 150;_parent._yscale = 150;  
}
```

You can reuse this script by attaching it to any movie clip instance.

Note: *Flash Lite 1.0 and 1.1 support attaching scripts only to buttons. Attaching scripts to movie clips is not supported.*

Whether you use an absolute or a relative path, you identify a variable in a Timeline or a property of an object with a dot (.) followed by the name of the variable or property. For example, the following statement sets the variable `name` in the instance `form` to the value "Gilbert":

```
_root.form.name = "Gilbert";
```

Using absolute and relative target paths

You can use ActionScript to send messages from one timeline to another. The timeline that contains the action is called the *controlling timeline*, and the timeline that receives the action is called the *target timeline*. For example, there could be an action on the last frame of one timeline that tells another timeline to play. To refer to a target timeline, you must use a target path, which indicates the location of a movie clip in the display list.

The following example shows the hierarchy of a document named `westCoast` on level 0, which contains three movie clips: `california`, `oregon`, and `washington`. Each of these movie clips in turn contains two movie clips.

```
_level0  
  westCoast  
    california  
      sanfrancisco  
      bakersfield  
    oregon  
      portland  
      ashland  
    washington  
      olympia  
      ellensburg
```

As on a web server, each timeline in Flash can be addressed in two ways: with an absolute path or with a relative path. The absolute path of an instance is always a full path from a level name, regardless of which timeline calls the action; for example, the absolute path to the instance `california` is `_level0.westCoast.california`. A relative path is different when called from different locations; for example, the relative path to `california` from `sanfrancisco` is `_parent`, but from `portland`, it's `_parent._parent.california`.

See also

[“Structuring FLA files”](#) on page 418

[“Organizing ActionScript in an application”](#) on page 420

Specify target paths

To control a movie clip, loaded SWF file, or button, you must specify a target path. You can specify it manually, or by using the Insert Target Path dialog box, or by creating an expression that evaluates to a target path. To specify a target path for a movie clip or button, you must assign an instance name to the movie clip or button. A loaded document doesn't require an instance name, because you use its level number as an instance name (for example, `_level5`).


Assign an instance name to a movie clip or button

- 1 Select a movie clip or button on the Stage.
- 2 Enter an instance name in the Property inspector.

Specify a target path using the Insert Target Path dialog box

- 1 Select the movie clip, frame, or button instance to which you want to assign the action.

This becomes the controlling Timeline.

- 2 In the Actions panel (Window > Actions), go to the Actions toolbox on the left, and select an action or method that requires a target path.
- 3 Click the parameter box or location in the script where you want to insert the target path.
- 4 Click the Insert Target Path button  above the Script pane.
- 5 Select Absolute or Relative for the target path mode.
- 6 Select a movie clip in the Insert Target Path display list, and click OK.

Specify a target path manually

- 1 Select the movie clip, frame, or button instance to which you want to assign the action.

This becomes the controlling Timeline.

- 2 In the Actions panel (Window > Actions), go to the Actions toolbox on the left, and select an action or method that requires a target path.
- 3 Click the parameter box or location in the script where you want to insert the target path.
- 4 Enter an absolute or relative target path in the Actions panel.

Use an expression as a target path

- 1 Select the movie clip, frame, or button instance to which you want to assign the action.

This becomes the controlling Timeline.

- 2 In the Actions panel (Window > Actions), go to the Actions toolbox on the left, and select an action or method that requires a target path.
- 3 Do one of the following:
 - Enter an expression that evaluates to a target path in a parameter box.
 - Click to place the insertion point in the script. Then, in the Functions category of the Actions toolbox, double-click the `targetPath` function. The `targetPath` function converts a reference to a movie clip into a string.
 - Click to place the insertion point in the script. Then, in the Functions category of the Actions toolbox, select the `eval` function. The `eval` function converts a string to a movie clip reference that can be used to call methods such as `play`.

The following script assigns the value 1 to the variable `i`. It then uses the `eval` function to create a reference to a movie clip instance and assigns it to the variable `x`. The variable `x` is now a reference to a movie clip instance and can call the `MovieClip` object methods.

```
i = 1;
x = eval("mc"+i);
x.play();
// this is equivalent to mc1.play();
```

You can also use the `eval` function to call methods directly, as shown in the following example:

```
eval("mc" + i).play();
```

Animation basics

Types of animation

Adobe® Flash® CS4 Professional provides several ways to create animation and special effects. Each method provides you with different possibilities for creating engaging animated content.

Flash supports the following types of animation:

Motion tweens Use motion tweens to set properties for an object, such as position and alpha transparency in one frame and again in another frame. Flash then interpolates the property values of the frames in between. Motion tweens are useful for animation that consists of continuous motion or transformation of an object. Motion tweens appear in the Timeline as a contiguous span of frames that can be selected as a single object by default. Motion tweens are powerful and simple to create.

Classic tweens Classic tweens are like motion tweens, but are more complex to create. Classic tweens allow for some specific animated effects not possible with span-based tweens.

Inverse Kinematics poses Inverse kinematic poses allow you to stretch and bend shape objects and link groups of symbol instances to make them move together in naturalistic ways. You can position the shape object or linked instances in different ways in separate frames, and Flash interpolates the positions in the frames in between.

Shape tweens In shape tweening, you draw a shape at one specific frame in the Timeline, and change that shape or draw another shape at another specific frame. Flash then interpolates the intermediate shapes for the frames in between, creating the animation of one shape morphing into another.

Frame-by-frame animation This animation technique lets you specify different art for each frame in the Timeline. Use this technique to create an effect that is like the frames of a film being played in rapid succession. This technique is useful for complex animation where the graphic elements of each frame must be different.

See also

[“Motion tweens”](#) on page 187

[“Working with classic tweens”](#) on page 210

[“Using inverse kinematics”](#) on page 223

[“Shape tweening”](#) on page 218

[“Frame-by-frame animation”](#) on page 220

About frame rates

The frame rate, the speed the animation is played at, is measured in number of frames per second (fps). A frame rate that's too slow makes the animation appear to stop and start; a frame rate that's too fast blurs the details of the animation. A frame rate of 24 fps is the default for new Flash documents and usually gives the best results on the web. The standard motion-picture rate is also 24 fps.

The complexity of the animation and the speed of the computer playing the animation affect the smoothness of the playback. To determine optimum frame rates, test your animations on a variety of computers.

Because you specify only one frame rate for the entire Flash document, set this rate before you begin creating animation.

See also

[“Create a new document”](#) on page 43

Representing animations in the Timeline

Flash distinguishes tweened animation from frame-by-frame animation in the Timeline by displaying different indicators in each frame that contains content.

The following frame content indicators appear in the Timeline:

- A span of frames with a blue background indicates a *motion tween*. A black dot in the first frame of the span indicates that the tween span has a target object assigned to it. Black diamonds indicate the last frame and any other property keyframes. Property keyframes are frames that contain property changes explicitly defined by you. You can choose which types of property keyframes to display by right-clicking (Windows) or Command-clicking (Macintosh) the motion tween span and choosing View Keyframes > *type* from the context menu. Flash displays all types of property keyframes by default. All other frames in the span contain interpolated values for the tweened properties of the target object.



- A hollow dot in the first frame indicates that the target object of the motion tween has been removed. The tween span still contains its property keyframes and can have a new target object applied to it.



- A span of frames with a green background indicates an inverse kinematics (IK) *pose layer*. Pose layers contain IK armatures and *poses*. Each pose appears in the Timeline as a black diamond. Flash interpolates the positions of the armature in the frames in between poses.



- A black dot at the beginning keyframe with a black arrow and blue background indicates a *classic tween*.



- A dashed line indicates that the classic tween is broken or incomplete, such as when the final keyframe is missing.



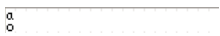
- A black dot at the beginning keyframe with a black arrow and a light green background indicates a *shape tween*.



- A black dot indicates a single keyframe. Light gray frames after a single keyframe contain the same content with no changes. These frames have a vertical black line and a hollow rectangle at the last frame of the span.



- A small *a* indicates that the frame is assigned a frame action with the Actions panel.



- A red flag indicates that the frame contains a label.



- A green double slash indicates that the frame contains a comment.



- A gold anchor indicates that the frame is a named anchor.



About layers in animation

Each scene in a Flash document can consist of any number of Timeline layers. Use layers and layer folders to organize the contents of an animation sequence and separate animated objects. Organizing them in layers and folders prevents them from erasing, connecting to, or segmenting each other when they overlap. To create animation that includes tweened movement of more than one symbol or text field at once, place each object on a separate layer. You can use one layer as a background layer to contain static artwork and use additional layers to contain one separate animated object.

When you create a motion tween, Flash converts the layer containing the object you selected to tween into a tween layer. The tween layer has a tween icon next to the layer name in the Timeline.

If other objects are present on the same layer as the tweened object, Flash adds new layers above or below the original layer as needed. Any objects that existed below the tweened object on its original layer move to a new layer below the original layer. Any objects that were above the tweened object on its original layer move to a new layer above the original layer. Flash inserts these new layers between any pre-existing layers in the Timeline. In this way Flash preserves the original stacking order of all the graphic objects on the Stage.

A tween layer can contain only tween spans (contiguous groups of frames containing a tween), static frames, blank keyframes, or empty frames. Each tween span can contain only a single target object and an optional motion path for the target object. Because you cannot draw in a tween layer, create additional tweens or static frames on other layers and then drag them to the tween layer. To place frame scripts on a tween layer, create them on another layer and drag them to the tween layer. A frame script can only reside in a frame outside the motion tween span itself. In general, it is best to keep all frame scripts on a separate layer that contains only ActionScript.

When a document has several layers, tracking and editing the objects on one or more of them can be difficult. This task is easier if you work with the contents of one layer at a time. To hide or lock layers you are not currently working on, click the Eye or Lock icon next to the layer name in the Timeline. Using layer folders can help you organize layers into manageable groups.

Distributing objects to layers for tweened animation

Flash automatically moves an object to its own tween layer when you apply a motion tween to the object. However, you can also distribute objects to their own separate layers yourself. For example, you can choose to distribute objects yourself when you are organizing content. Manual distribution is also useful for applying animation to objects while maintaining precise control over how they move from one layer to another.

When you use the Distribute To Layers command (Modify > Timeline > Distribute To Layers), Flash distributes each selected object to a new, separate layer. Any objects that you don't select (including objects in other frames) are preserved in their original layers.

You can apply the Distribute To Layers command to any element on the Stage, including graphic objects, instances, bitmaps, video clips, and broken-apart text blocks.

See also

[“About layers in animation”](#) on page 185

[“Break text apart”](#) on page 252

[“Motion tweens”](#) on page 187

About new layers created with Distribute to Layers

New layers created during the Distribute To Layers operation are named according to the name of the element that each contains:

- A new layer containing a library asset (such as a symbol, bitmap, or video clip) receives the same name as the asset.
- A new layer containing a named instance receives the name of the instance.
- A new layer containing a character from a broken-apart text block is named with the character.
- A new layer containing a graphic object (which has no name) is named Layer1 (or Layer2, and so on), because graphic objects do not have names.

Flash inserts the new layers below any selected layers. The new layers are arranged top to bottom, in the order that the selected elements were originally created. The layers in broken-apart text are arranged in the order of the characters, whether left-to-right, right-to-left, or top-to-bottom. For example, suppose you break apart the text *FLASH* and distribute it to layers. The new layers, named F, L, A, S, and H, are arranged top to bottom, with F at the top. These layers appear immediately below the layer that initially contained the text.

Distribute objects to layers

- 1 Select the objects you want to distribute to separate layers. The objects can be in a single layer, or in several layers, including non-contiguous layers.
- 2 Do one of the following:
 - Select Modify > Timeline > Distribute To Layers.
 - Right-click (Windows) or Control-click (Macintosh) one of the selected objects, and select Distribute To Layers.

Additional resources

Jen DeHaan has written a detailed article titled [Creating animation in ActionScript 3.0](#), available in the Adobe Flash Professional Developer Center.

Motion tweens

About tweened animation

A *tween* is an animation that is created by specifying a value for an object property in one frame and another value for that same property in another frame. Flash calculates the values for that property in between those two frames. The term tween comes from the words “in between”.

For example, you can place a movie clip on the left side of the Stage in frame 1 of the Timeline, and then move that movie clip to the right side of the Stage in frame 20. When you create a tween, Flash calculates all the positions of the movie clip on Stage in between the two positions, right and left, that you specified. You end up with an animation of the movie clip moving from the left side of the Stage to the right side, from frame 1 to frame 20. In each frame in between, Flash moves the movie clip one 20th of the distance across the Stage.

The types of objects that can be tweened include movie clip, graphic and button symbols, and text fields. The properties of these objects that can be tweened include the following:

- 2D X and Y position
- 3D Z position (movie clips only)
- 2D rotation (around the z axis)
- 3D X, Y, and Z rotation (movie clips only)

3D motion requires that the FLA file target ActionScript 3.0 and Flash Player 10 in the publish settings.

- Skew X and Y
- Scale X and Y
- Color effects

Color effects include alpha (transparency), brightness, tint, and advanced color settings. Color effects can be tweened only on symbols. To tween a color effect on text, convert the text to a symbol.

- Filter properties (not including filters applied to graphic symbols)

A *tween span* is a group of frames in the Timeline in which an object on the Stage can have one or more properties changed over time. A tween span appears in the Timeline as a group of frames in a single layer with a blue background. These tween spans can be selected as a single object and dragged from one location in the Timeline to another, including to another layer. Only one object on the Stage can be animated in each tween span. This object is called the *target object* of the tween span.

A *property keyframe* is a frame within a tween span where you explicitly define one or more property values for the tween target object. Each property you define has its own property keyframes. If you set more than one property in a single frame, then the property keyframes for each of those properties reside in that frame. You can view each property of a tween span and its property keyframes in the Motion Editor. You can also choose which types of property keyframes to display in the Timeline from the tween span context menu.

In the preceding example of tweening a movie clip from the left side of the Stage to the right side from frame 1 to frame 20, frames 1 and 20 are property keyframes. You can use the Property inspector, the Motion Editor, the Transform panel, or various tools in the Tools panel to define explicit values for properties you want to animate. You specify these property values in the frames of your choosing, and Flash adds the required property keyframes to the tween span. Flash interpolates the values for each of these properties in the frames in between the property keyframes you have created.

Note: Beginning in Flash Professional CS4, the concepts of a “keyframe” and a “property keyframe” are different. The term *keyframe* refers to a frame in the Timeline in which a symbol instance appears on the Stage for the first time. The separate term *property keyframe*, which is new to Flash CS4, refers to a value defined for a property at a specific time or frame in a motion tween.

If a tweened object changes its location on the Stage during the tween, the tween span has a motion path associated with it. This motion path shows the path the tweened object takes during its movement around the Stage. You can edit the motion path on the Stage using the Subselection, Convert Anchor, Remove Anchor, and Free Transform tools, and commands in the Modify menu. If you are not tweening position, no motion path appears on the Stage. You can also apply an existing path as a motion path by pasting the path onto a tween span in the Timeline.

Tweened animation is an effective and efficient way to create movement and changes over time while minimizing file size. In tweened animation, only the values for the property keyframes you specify are stored in the FLA file and published SWF file.

Before creating tweens, it is helpful to understand the following Flash concepts:

- Drawing on the Stage
- Timeline layers and the stacking order of objects within a single layer as well as across layers
- Moving and transforming objects on the Stage and in the Property inspector
- Using the Timeline, including object lifetime and selecting objects at specific points in time
- Symbols. Tweenable symbol types include movie clips, buttons, graphics, and text.
- Optional: Bezier curve editing using the Select and Subselect tools. These tools can be used for editing tween motion paths.

For more information about these concepts, refer to the list of links at the bottom of this page.

For video tutorials about animation, see the following:

- Creating motion tweens: www.adobe.com/go/lrvid4054_fl
- Creating shape tweens: www.adobe.com/go/vid0124
- Creating and animating masks: www.adobe.com/go/vid0127

See also

“Drawing” on page 91

“Create and organize layers” on page 173

“Transforming objects” on page 113

“Working with timelines” on page 172

“Working with symbols” on page 147

“Reshape lines and shapes” on page 111

“Easing tweens” on page 207

“Shape tweening” on page 218

“Break text apart” on page 252

“Break apart groups and objects” on page 122

Differences between motion tweens and classic tweens

Flash supports two different types of tweens for creating motion. *Motion tweens*, introduced in Flash CS4 Professional, are powerful and simple to create. Motion tweens allow the greatest control over tweened animation. *Classic tweens*, which include all tweens created in earlier versions of Flash, are more complex to create. While motion tweens offer much more control of a tween, classic tweens provide certain specific capabilities that some users may want.

The differences between motion tweens and classic tweens include the following:

- Classic tweens use keyframes. Keyframes are frames in which a new instance of an object appears. Motion tweens can only have one object instance associated with them and use property keyframes instead of keyframes.
- A motion tween consists of one target object over the entire tween span.
- Both motion tweens and classic tweens allow only specific types of objects to be tweened. Applying a motion tween will convert all non-allowed object types to a movie clip when the tween is created. Applying a classic tween will convert them to graphic symbols.
- Motion tweens consider text a tweenable type and do not convert text objects to movie clips. Classic tweens convert text objects to graphic symbols.
- No frame scripts are allowed on a motion tween span. Classic tweens allow frame scripts.
- Any object scripts on a tween target cannot change over the course of the motion tween span.
- Motion tween spans can be stretched and resized in the Timeline and are treated as a single object. Classic tweens consist of groups of individually selectable frames in the Timeline.
- To select individual frames in a motion tween span you must Ctrl-click (Windows) or Command-click (Macintosh) the frames.
- With classic tweens, eases can be applied to the groups of frames between the keyframes within the tween. With motion tweens, eases apply across the entire length of a motion tween span. Easing only specific frames of a motion tween requires the creation of a custom ease curve.
- You can use classic tweens to animate between two different color effects, such as tint and alpha transparency. Motion tweens can apply one color effect per tween.
- Only motion tweens can be used to animate 3D objects. You cannot animate a 3D object using a classic tween.
- Only motion tweens can be saved as motion presets.
- With motion tweens, you cannot swap symbols or set the frame number of a graphic symbol to display in a property keyframe. Animations that include these techniques require classic tweens.

Jen DeHaan provides a useful blog post about the motion model in Flash CS4 and the differences between motion tweens and classic tweens on her Flashthusiast.com site.

Applying motion presets

Motion presets are pre-configured motion tweens that you can apply to an object on the Stage. You simply select the object and click the Apply button in the Motion Presets panel.

Using motion presets is a quick way to learn the basics of adding animation in Flash. Once you see how presets work, making your own animations is even easier.

You can create and save your own custom presets. These can be from existing motion presets that you have modified, or from custom tweens that you have created on your own.

The Motion Presets panel also allows you to import and export presets. You can share presets with people you are collaborating with or take advantage of presets shared by members of the Flash design community.

Using presets can save significant production time during design and development of your projects, especially if you use similar kinds of tweens often.

Note: Motion presets can only contain motion tweens. Classic tweens cannot be saved as motion presets.

For a video tutorial about motion presets, see Using Motion Presets at www.adobe.com/go/lrvid4056_fl.

See also

“Create a motion tween” on page 193

“Edit the motion path of a tween” on page 195

Preview a motion preset

Each motion preset included with Flash has a preview that you can watch in the Motion Presets panel. The preview gives you a sense of what the animation looks like when applied to an object in your FLA file. For custom presets that you create or import, you can add your own preview.

- 1 Open the Motion Presets panel.
- 2 Select a motion preset from the list.

The preview plays in the Preview pane at the top of the panel.

- 3 To stop the preview from playing, click outside the Motion Presets panel.

Apply a motion preset

While a tweenable object (symbol instance or text field) is selected on the Stage, you can click the Apply button to apply a preset. You can apply only one preset per object. If you apply the second preset to the same object, the second preset replaces with the first one.

Once a preset is applied to an object on the Stage, the tween created in the Timeline no longer has any relationship to the Motion Presets panel. Deleting or renaming a preset in the Motion Presets panel has no effect on any tweens previously created with that preset. If you save a new preset over an existing one in the panel, it has no effect on any tweens already created with the original preset.

Each motion preset contains a specific number of frames. When you apply a preset, the tween span created in the Timeline contains this number of frames. If the target object already had a tween of a different length applied to it, the tween span adjusts to match the length of the motion preset. You can adjust the length of the tween span in the Timeline after the preset is applied.

Motion presets that contain 3D motion can only be applied to movie clip instances. The tweened 3D properties do not apply to graphic or button symbols, or to text fields. You can apply 2D or 3D motion presets to any 2D or 3D movie clip.

Note: Motion presets that animate the z axis position of a 3D movie clip will cause the movie clip to appear to change its x and y position as well. This is because movement along the z axis follows the invisible perspective lines which radiate from the 3D vanishing point (set in the 3D symbol instance Property inspector) to the edges of the Stage.

To apply the motion preset:

- 1 Select a tweenable object on the Stage. If you apply a motion preset to an object that is not tweenable, a dialog box appears to allow you to convert the object to a symbol.
- 2 Select a preset in the Motion Presets panel.
- 3 Click the Apply button in the panel or choose Apply at Current Location from the panel menu.

The motion is applied so that the motion starts at the current position of the movie clip on the Stage. If the preset has a motion path associated with it, the motion path appears on the Stage.

To apply the preset so that its motion ends at the current position of the object on the Stage, Shift-click the Apply button or select End at Current Location from the panel menu.

You can also apply a motion preset to multiple selected frames on separate layers, as long as each selected frame contains only a single tweenable object.

Save a tween as a custom motion preset

If you create your own tween or change a tween that you applied from the Motion Presets panel, you can save it as a new motion preset. The new preset appears in the Custom Presets folder in the Motion Presets panel.

To save a custom tween as a preset:

- 1 Select one of the following items:
 - The tween span in the Timeline
 - The object on Stage to which the custom tween was applied
 - The motion path on the Stage
- 2 Click the Save Selection As Preset button in Motion Presets panel or choose Save as Motion Preset from the context menu of the selection.

The new preset appears in the Motion Presets panel. Flash saves the preset as an XML file. The files are stored in the following directories:

- Windows: <hard disk>\Documents and Settings\<user>\Local Settings\Application Data\Adobe\Flash CS4\<language>\Configuration\Motion Presets\
- Macintosh: <hard disk>/Users/<user>/Library/Application Support/Adobe/Flash CS4/<language>/Configuration/Motion Presets/

Note: You cannot undo saving, deleting, or renaming custom presets.

See also

[“Create motion tweens”](#) on page 192

Import a motion preset

Motion presets are stored as XML files. Import an XML tween file to add it to the Motion Presets panel.

- 1 Choose Import from the Motion Presets panel menu.
- 2 In the Open dialog box, navigate to the XML file you want to import and click Open.

Flash opens the XML file and adds the motion preset to the panel.

Export a motion preset

You can export motion presets as XML files that can be shared with other Flash users.

- 1 Select the preset in the Motion Presets panel.
- 2 Choose Export from the panel menu.
- 3 In the Save As dialog box, choose a name and location for the XML file and click Save.

Delete a motion preset

You can remove presets from the Motion Presets panel. When you remove a preset, Flash deletes its XML file from disk. Consider making a backup of any presets you might want to use again later by exporting copies of them first.

- 1 Select the preset to delete in the Motion Presets panel.
- 2 Do one of the following:
 - Choose Remove from the panel menu.
 - Click the Remove item button in the panel.

Create a preview for a custom preset

You can create a preview for any custom motion presets that you create. Do this by storing a SWF file that demonstrates the tweened animation in the same directory with the motion preset XML file.

- 1 Create the tweened animation and save it as a custom preset.
- 2 Create a FLA file that contains only a demonstration of the tween. Save the FLA with the exact same name as the custom preset.
- 3 Create a SWF file from the FLA file with the Publish command.
- 4 Place the SWF file in the same directory as the saved custom motion preset XML file. These files are stored in the following directories:
 - Windows: <hard disk>\Documents and Settings\<user>\Local Settings\Application Data\Adobe\Flash CS4\<language>\Configuration\Motion Presets\
 - Macintosh: <hard disk>/Users/<user>/Library/Application Support/Adobe/Flash CS4/<language>/Configuration/Motion Presets/

The preview now displays when the custom tween is selected in the Motion Presets panel.

Create motion tweens

Tweens are applied to symbol instances and text fields. Only symbol instances and text fields can be tweened. All other object types are wrapped in a symbol when a tween is applied to them. The symbol instance can contain nested symbols, which can themselves be tweened on their own timelines.

The minimal building block in a tween layer is a tween span. A tween span in a tween layer can contain only one symbol instance. The symbol instance is called the target instance of the tween span. Adding a second symbol to the tween span replaces the original symbol in the tween. You can change the target object of a tween by dragging a different symbol from the library onto the tween span in the Timeline. You can delete the symbol from a tween layer without removing or breaking the tween. You can then add a different symbol instance to the tween at a later time. You can also change the type of the target symbol of a tween span.

You can edit individual property keyframes on Stage, in the Property inspector, or in the Motion Editor.

When you add a tween to an object or set of objects on a layer, Flash either converts the layer to a tween layer or creates a new layers to preserve the original stacking order of the objects on the layer, according to these rules:

- If there are no objects on the layer other than the selection, the layer changes to a tween layer.
- If the selection is at the bottom of the stacking order of the layer (under all other objects), a layer will be created above the original layer to hold the non-selected items and the original layer becomes a tween layer.
- If the selection is at the top of the stacking order of the layer (above all other objects), a new layer is created, the selection is moved to it and that layer becomes a tween layer.

- If the selection is in the middle of the stacking order of the layer (there are non-selected objects above and below the selection), two layers are created, one to hold the new tween and one above it to hold the non-selected items at the top of the stacking order. The non-selected items at the bottom of the stacking order remain on the original layer, below the newly inserted layers.

A tween layer can contain tween spans as well as static frames and ActionScript. However, frames of a tween layer that contain a tween span cannot contain objects other than the tweened object. To add additional objects in the same frame, place them on separate layers.

When a tween contains motion, a motion path appears on the Stage. The motion path shows the position of the tweened object in each frame. You cannot add a motion guide to a tween/inverse kinematics layer.

For a video tutorial about motion tweens, see *Creating Motion Tweens, Part 1* at www.adobe.com/go/lrvid4054_fl and *Creating Motion Tweens, Part 2* at www.adobe.com/go/lrvid4055_fl.

For information about tweening Inverse Kinematic bones, see “[Animate an armature](#)” on page 230.

See also

“[Editing property curves with the Motion Editor](#)” on page 203

“[Animate an armature](#)” on page 230

“[3D graphics](#)” on page 139

Create a motion tween

- 1 Select one or more objects to tween on the Stage.

The object can reside in any of the following layer types: Normal, Guide, Mask, or Masked.

- 2 Do one of the following:

- Choose Insert > Motion Tween.
- Right-click (Windows) or Ctrl-click (Macintosh) the selection or current frame and choose Create Motion Tween from the context menu.

If the object is not a tweenable object type, or if multiple objects are selected on the same layer, a dialog box appears. The dialog box allows you to convert the selection to a movie clip symbol. Convert the selection to a movie clip to proceed.

If the tweened object was the only item on the layer, Flash converts the layer containing the object to a tween layer. If there are other objects on the layer, Flash inserts layers to preserve the original object stacking order and places the tweened object on its own layer.

If the original object resided in only the first frame of the Timeline, the length of the tween span is equal to one second in duration. If the frame rate is 24 frames per second, the span contains 24 frames. If the frame rate is less than five fps, the span is five frames long. If the original object was present in more than one contiguous frame, the tween span contains the number of frames occupied by the original object.

If the layer was a normal layer it becomes a tween layer. If it was a guide, mask, or masked layer it becomes a tween guide, tween mask, or tween masked layer.

- 3 Drag either end of the tween span in the Timeline to shorten or extend the span to the desired length.
- 4 To add motion to the tween, place the playhead on a frame within the tween span and then drag the object on the Stage to a new position.

A motion path appears on the Stage showing the path from the position in the first frame of the tween span to the new position. Because you explicitly defined the X and Y properties of the object, property keyframes are added for X and Y in the frame containing the playhead. Property keyframes appear as small diamonds in the tween span.

By default, the Timeline displays the property keyframes of all property types. You can choose which types of property keyframes to display by right-clicking (Windows) or Command-clicking (Macintosh) the tween span and choosing View Keyframes > *property type*.

- 5 To specify another position for the object, place the playhead in another frame within the tween span and drag the object on Stage to another position.

The motion path adjusts to include all the positions you specify.

- 6 To tween 3D rotation or position, use the 3D Rotation or 3D Translation tool. Be sure to place the playhead in the frame where you want to add the 3D property keyframe first.

Note: To create multiple tweens at once, place tweenable objects on multiple layers, select them all, and choose Insert > Motion Tween. You can also apply motion presets to multiple objects in the same way.

Tween a non-position property with the Property inspector

Use the Property inspector to edit the value of any property on tween in the current frame.

- 1 Select an object on the Stage.
- 2 Choose Insert > Motion Tween.

If the object must be converted to a symbol, click OK in the dialog box that appears. Flash converts the object to a movie clip symbol.

When you apply a tween to an object that exists only in a single keyframe, the playhead moves to the last frame of the new tween. Otherwise the playhead does not move.

- 3 Place the playhead in the frame of the tween span where you want to specify a property value.

You can place the playhead in any other frame of the tween span. The tween starts with the property values in the first frame of the tween span, which is always a property keyframe.

- 4 With the object selected on the Stage, set a value for a non-position property such as alpha transparency, skew, and so on. Set the value with the Property inspector or with one of the tools in the Tools panel.

The current frame of the span becomes a property keyframe.

To display different types of property keyframes in tween spans, right-click (Windows) or Ctrl-click (Macintosh) a tween span and choose View Keyframes > *property type* from the context menu.

- 5 Scrub the playhead in the Timeline to observe the tween on the Stage.
- 6 To add additional property keyframes, move the playhead to the desired frame in the span and set a value for the property in the Property inspector.

Note: You can also tween properties along an entire tween with the Motion Editor. See [“Editing property curves with the Motion Editor”](#) on page 203 for more information.

Add an additional tween to an existing tween layer

You can add additional tweens to an existing tween layer.

- ❖ Do one of the following:
 - Add a blank keyframe to the layer (Insert > Timeline > Blank Keyframe), add items to the keyframe, and then tween one or more of those items.

- Create a tween on a separate layer and then drag the span to the desired layer.
- Drag a static frame from another layer to the tween layer and then add a tween to an object in the static frame.
- Insert a blank keyframe on the tween layer and then add an object to the blank keyframe by dragging one from the Library panel or pasting from the clipboard. You can then add a tween to this object.

Note: You can copy the target object of a motion tween to the clipboard at any frame in the tween span.

Edit the motion path of a tween

You can edit the motion path of a tween in the following ways:

- Change the position of the object in any frame of the tween span.
- Move the entire motion path to a different location on the Stage.
- Change the shape or size of the path with the Selection, Subselection, or Free Transform tools.
- Change the shape or size of the path with the Transform panel or Property inspector.
- Use the commands in the Modify > Transform menu.
- Apply a custom stroke as a motion path.
- Use the Motion Editor.

You can use the Always Show Motion Paths option to show all motion paths on all layers on the Stage simultaneously. This display is helpful when designing multiple animations on different motion paths that intersect each other. When a motion path or tween span is selected, you can choose this option from the Property inspector options menu.

See also

[“Editing property curves with the Motion Editor”](#) on page 203

Change the position of the tweened object

The simplest way to edit a motion path is to move the target instance of the tween in any frame of the tween span. If the frame does not already contain a property keyframe, Flash adds one to it.

- 1 Place the playhead in the frame where you want to move the target instance.
- 2 With the Selection tool, drag the target instance to a new location on the Stage.

The motion path updates to include the new location. All other property keyframes in the motion path remain in their original locations.

Change the location of a motion path on the Stage

You can drag the entire motion path on the Stage or set its location in the Property inspector.

- 1 Click the Selection tool from Tools panel.
- 2 Select the motion path by doing one of the following:
 - Click the motion path. Flash displays the Motion Tween properties in the Property inspector. Do not click the tween target instance.
 - Drag a marquee around the motion path and the target instance to select them both.
- 3 Move the motion path by doing one of the following:
 - Drag the path to desired location on the Stage.

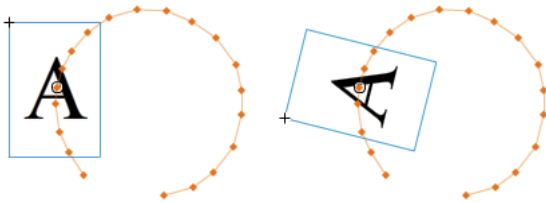
- Set the X and Y values for the path in the Property inspector. The X and Y values are for the upper-left corner of the bounding box of the motion path.
- Use the Arrow keys to move the motion path.

Note: To move the tween target instance and the motion path by specifying a location for the motion path, select both and enter the X and Y locations in the Property inspector. To move a tweened object that has no motion path, select it and enter X and Y values in the Property inspector.

Edit the shape of a motion path with the Selection and Subselection tools

With the Selection and Subselection tools, you can reshape a motion path. With the Selection tool, you can reshape a segment by dragging. Property keyframes in the tween appear on the path as control points. With the Subselection tool, you can expose the control points and Bezier handles on the path that correspond to each position property keyframe. You can use these handles to reshape the path around the property keyframe points.

When you create a non-linear motion path such as a circle, you can have the tweened object rotate as it moves along the path. To maintain a constant orientation relative to the path, select the Orient to Path option in the Property inspector.



A tweened object not oriented to the motion path (left) and oriented to the motion path (right).

- 1 Click the Selection tool in the Tools panel.
- 2 Click the Stage away from the motion path and the tween target instance so that neither is selected.
- 3 With the Selection tool, drag any segment of the motion path to reshape it. Do not click to select the segment first.
- 4 To expose the Bezier control points of a property keyframe point on the path, click the Subselection tool and then click the path.

The property keyframe points appear as control points (small diamonds) on the motion path.

- 5 To move a control point, drag it with the Subselection tool.
- 6 To adjust the curve of the path around a control point, drag the Bezier handles of the control point with the Subselection tool.

If the handles are not extended, you can extend them by Alt-dragging (Windows) or Option-dragging (Macintosh) the control point.

Edit a motion path with the Free Transform tool

- 1 Click the Free Transform tool in the Tools panel.
- 2 With the Free Transform tool, click the motion path. Do not click the tween target instance.
- 3 Scale, skew, or rotate the path with the Free Transform tool.

Note: You can also perform free transforms on the path by selecting it with the Subselection tool and then pressing the Control (Windows) or Command (Macintosh) key. Pressing the key displays the same controls as the Free Transform tool. You can then drag to perform transforms while pressing the key.

Delete a motion path from a tween

- 1 Select the motion path on the Stage by clicking it with the Selection tool.
- 2 Press the Delete key.

Copy a motion path as a stroke

- 1 Click the motion path on the Stage to select it.
- 2 Choose Edit > Copy.

You can then paste the path into another layer as a stroke or as a motion path for another motion tween.

Apply a custom stroke as a motion path

You can apply a stroke from a separate layer or a separate timeline as the motion path for a tween.

- 1 Select a stroke on a layer separate from the tween layer and copy it to the clipboard.

The stroke must not be closed. Only uninterrupted strokes can be used.

- 2 Select a tween span in Timeline.
- 3 With the tween span still selected, paste the stroke.

Flash applies the stroke as the new motion path for the selected tween span. The target instance of the tween now moves along the new stroke.

- 4 To reverse the start and end points of the tween, right-click (Windows) or Ctrl-click (Macintosh) the tween span and select Motion Path > Reverse Path in the tween span context menu.

Use roving property keyframes

A roving property keyframe is a keyframe that is not linked to a specific frame in the Timeline. Flash adjusts the position of roving keyframes so that the speed of motion is consistent throughout a tween.

Roving keyframes are available only for the spatial properties X, Y, and Z. They are useful when you have edited a motion path on the Stage by dragging the tweened object to different locations in different frames. Editing motion paths this way often creates path segments in which the motion is faster or slower than the other segments. This is because the number of frames in the path segment is greater or smaller than other segments.

Using roving property keyframes is helpful for making the speed of an animation consistent throughout a tween. When property keyframes are set to roving, Flash adjusts the position of the property keyframes in the tween span so that the tweened object moves the same distance in each frame of the tween. You can then use easing to adjust the movement so that the acceleration at the beginning and end of the tween has a realistic appearance.

When you paste a custom path onto a tween, Flash sets the property keyframes to roving by default.

To enable roving keyframes for an entire tween:

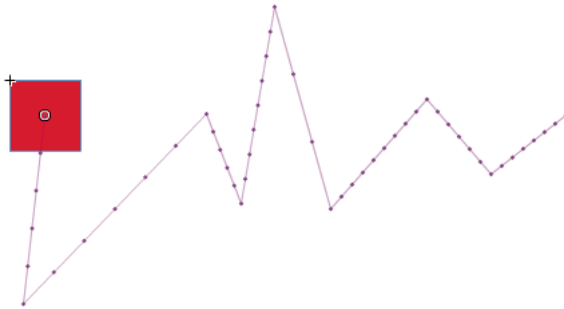
- Right-click (Windows) or Command-click (Macintosh) the tween span in the Timeline and choose Motion Path > Switch keyframes to roving in the context menu.

To enable roving for an individual property keyframe in a tween:

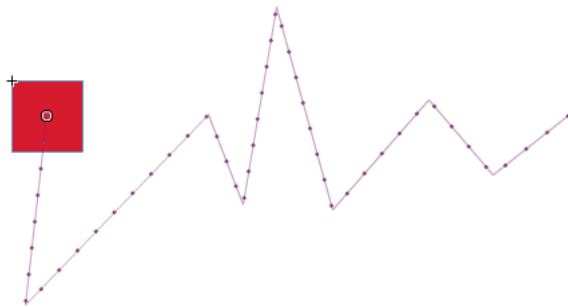
- Right-click (Windows) or Command-click (Macintosh) the property keyframe in the Motion Editor panel and choose roving in the context menu. For more information about the Motion Editor, see [“Editing property curves with the Motion Editor”](#) on page 203.

When property keyframes are set to roving, they appear as round dots instead of squares in the Motion Editor.

Note: If you enable roving keyframes for a tween span and then disable them, the keyframes retain their locations in the span that resulted from enabling roving.



A motion path with roving keyframes turned off. Note the uneven distribution of frames, resulting in uneven speed of motion.



The same motion path with roving keyframes turned on, resulting in even distribution of frames along the path and even speed of motion.

Work with tween spans in the Timeline

When creating animation in Flash, it is often appropriate to set up tween spans in the Timeline first. By establishing the initial arrangement of objects in layers and frames, you can then finish the tweens by editing the tweened properties in the Property inspector or the Motion Editor.

To select tween spans and frames in the Timeline, do one of the following:

- To select an entire tween span, click the span.
- To select multiple tween spans, including non-contiguous spans, Shift-click each span.
- To select a single frame within a tween span, Ctrl-click (Windows) or Command-click (Macintosh) the frame inside the span.
- To select multiple contiguous frames within a span, Ctrl-drag (Windows) or Command-drag (Macintosh) inside the span.
- To select frames within multiple tween spans on different layers, Ctrl+drag across multiple layers.
- To select an individual property keyframe in a tween span, Ctrl+click the property keyframe. You can then drag it to a new location.

See also

[“Editing property curves with the Motion Editor”](#) on page 203

Move, duplicate, or delete tween spans

- To move a span to a new location in the same layer, drag the span.

Note: Locking a layer prevents editing on the Stage but not the Timeline. Moving a span on top of another span consumes the overlapped frames of the second span.

- To move a tween span to a different layer, drag the span to the layer or copy and paste the span into the new layer. You can drag a tween span onto an existing normal layer, tween layer, guide layer, mask layer, or masked layer. If the new layer is a normal empty layer, it becomes a tween layer.
- To duplicate a span, Alt-drag (Windows) or Command-drag (Macintosh) the span to a new location in the Timeline, or copy and paste the span.
- To delete a span, select the span and choose Remove Frames or Clear Frames from the span context menu.

Edit adjacent tween spans

- To move the breakline between two contiguous tween spans, drag the breakline.

Each tween is recalculated.

- To separate the adjacent start and end frames of two contiguous tween spans, Alt-drag (Windows) or Command-drag (Macintosh) the start frame of the second span.

This allows you to make room for frames between the two spans.

- To split a tween span into two separate spans, Ctrl-click (Windows) or Command-click (Macintosh) a single frame in the span and then choose Split Motion from the span context menu.

Both tween spans have same target instance.

Note: You cannot split motion if more than one frame is selected. If the tween that was split had easing applied, the two smaller tweens may not have exactly the same motion as the original.

- To join two contiguous tween spans, select both spans and choose Join Motions from the span context menu.

Edit the length of a tween span

- To change the length of an animation, drag the right or left edge of the tween span.

Dragging the edge of one span into the frames of another span replaces the frames of the second span.

- To extend the presence of a tweened object on Stage beyond either end of its tween, Shift-drag either end frame of its tween span. Flash adds frames to the end of the span without tweening those frames.

You can also select a frame after the tween span in the same layer and press F6. Flash extends the tween span and adds a property keyframe for all properties to the selected frame. If you press F5, Flash adds frames but will not add a property keyframe to the selected frame.

Note: To add static frames to the end of a span that is immediately adjacent to another span, first move the adjacent span to make room for the new frames.

Add or remove frames within a tween span

- To remove frames from within a span, Ctrl-drag (Windows) or Command-drag (Macintosh) to select frames and choose Remove Frames from the span context menu.
- To cut frames from within a span, Ctrl-drag (Windows) or Command-drag (Macintosh) to select the frames and then choose Cut Frames from the span context menu.

- To paste frames into an existing tween span, Ctrl-drag (Windows) or Command-drag (Macintosh) to select the frames to replace and choose Paste Frames from the span context menu.

Simply pasting an entire span onto another span replaces the entire second span.

Replace or remove the target instance of a tween

To replace the target instance of a tween span, do one of the following:

- Select the span and then drag the new symbol from the Library panel onto the Stage.
- Select the new symbol in the Library panel and the target instance of the tween on the Stage and choose Modify > Symbol > Swap Symbol.
- Select the span and paste a symbol instance or text from the clipboard.

To remove the target instance of a tween span without removing the tween, select the span and press the Delete key.

View and edit property keyframes of a tween span

- To view frames containing property keyframes in a span for different properties, select the span and choose View Keyframes from the span context menu and then choose the property type from the submenu.
- To remove a property keyframe from a span, Ctrl-click (Windows) or Command-click (Macintosh) the property keyframe to select it and then right-click (Windows) or Ctrl-click (Macintosh) the property keyframe and choose Clear Keyframe for the property type you want to delete the keyframe for.
- To add property keyframes for a specific property type to a span, Ctrl-click (Windows) or Command-click (Macintosh) to select one or more frames in the span. Then right-click (Windows) or Ctrl-click (Macintosh) and choose Insert Keyframe > *property type* from the span context menu. Flash adds property keyframes to the selected frames. You can also set a property of the target instance in a selected frame to add a property keyframe.
- To add a property keyframe for all property types to a span, place the playhead in the frame where you want to add the keyframe and choose Insert > Timeline > Keyframe, or press F6.
- To reverse the direction of motion of a tween, choose Motion Path > Reverse Path from the span context menu.
- To change a tween span to static frames, select the span and choose Remove Tween from the span context menu.
- To convert a tween span to a frame-by-frame animation, select the span and choose Convert to Frame by Frame Animation from the span context menu.
- To move a property keyframe to a different frame in the same tween span or a different tween span, Ctrl-click (Windows) or Command-click (Macintosh) the property keyframe to select it and then drag the property keyframe to the new location.
- To copy a property keyframe to another location in the tween span, Ctrl-click (Windows) or Command-click (Macintosh) the property keyframe to select it and then Alt-drag (Windows) or Option-drag (Macintosh) the property keyframe to the new location.

Add or remove 3D property keyframes in a tween

- ❖ Right-click (Windows) or Ctrl-click (Macintosh) the tween span in the Timeline and choose 3D tween from the context menu.

If the tween span contained no 3D property keyframes, Flash adds them to each existing X and Y position and rotation property keyframe. If the tween span already contained 3D property keyframes, Flash removes them.

Copy and paste a motion tween

You can copy the tweened properties from one tween span to another. The tweened properties are applied to the new target object, but the location of the target object is not changed. This allows you to apply a tween from one area of the Stage to an object in another area without repositioning the new target object.

- 1 Select the tween span that contains the tweened properties you want to copy.
- 2 Select Edit > Timeline > Copy Motion.
- 3 Select the tween span to receive the copied tween.
- 4 Select Edit > Timeline > Paste Motion.

Flash applies the tweened properties to the target tween span and adjusts the length of the tween span to match the copied tween span.

To copy a motion tween to the Actions panel or use it in another project as `ActionScript®`, use the Copy Motion as `ActionScript 3.0` command.

Copy and paste motion tween properties

You can copy the properties from a selected frame to another frame on the same tween span or a different tween span. The property values are added only to the selected frame when the properties are pasted. The copied property values of color effects, filters, and 3D properties are only pasted if the tweened object at that frame has the color effect, filter, or 3D properties already applied. 2D position properties cannot be pasted onto a 3D tween.

- 1 To select a single frame in a tween span, Ctrl-click (Windows) or Command-click (Macintosh) the frame.
- 2 Right-click (Windows) or Command-click (Macintosh) the selected frame and choose Copy Properties from the context menu.
- 3 Select a single frame to receive the copied properties by Ctrl-clicking (Windows) or Command-clicking (Macintosh) the frame.

The destination frame must be in a tween span.

- 4 To paste the copied properties into the selected frame, do one of the following:
 - To paste all of the copied properties, right-click (Windows) or Command-click (Macintosh) the selected frame of the target tween span and choose Paste Properties from the context menu.
 - To paste only some of the copied properties, right-click (Windows) or Command-click (Macintosh) the selected frame of the target tween span and choose Paste Properties Special from the context menu. In the dialog box that appears, select the properties to paste and click OK.

Flash creates a property keyframe for each of the pasted properties in the selected frame and reinterpolates the motion tween.

Convert a tween span to frame-by-frame animation

You can convert a motion tween span to frame-by-frame animation. In frame-by-frame animation, each frame contains separate keyframes (not property keyframes) which each contains separate instances of the animated symbol. Frame-by-frame animation does not contain interpolated property values. For more information, see [“Frame-by-frame animation”](#) on page 220.

- ❖ Right-click (Windows) or Ctrl-click (Macintosh) the tween span you want to convert and choose Convert to Frame by Frame Animation from the context menu.

Copy motion as ActionScript 3.0

Copy the properties that define a motion tween in the Timeline as ActionScript 3.0 and apply that motion to another symbol, either in the Actions panel or in the source files (such as class files) for a Flash document that uses ActionScript 3.0.

Use the `fl.motion` classes to customize the Flash-generated ActionScript for your specific project. For more information, see the `fl.motion` classes in the *ActionScript 3.0 Language and Components Reference*.

Copy Motion As ActionScript 3.0 can capture the following properties of a motion tween:

- Position
- Scale
- Skew
- Rotation
- Transformation Point
- Color
- Blend Mode
- Orientation To Path
- Cache As Bitmap Setting
- Easing
- Filters
- 3D rotation and position.

1 Select the tween span in the Timeline or the object on the Stage that contains the motion tween to copy.

Only one tween span or tweened object can be selected to copy as ActionScript 3.0.

2 Do one of the following:

- Select Edit > Timeline > Copy Motion As ActionScript 3.0.
- Right-click (Windows) or Control-click (Macintosh) the tween span or the tweened instance on the Stage and choose Copy Motion As ActionScript 3.0.

Flash copies to your system's clipboard the ActionScript 3.0 code that describes the selected motion tween. The code describes the tween as a frame-by-frame animation.

To use the copied code, paste it into the Actions panel of a Flash document that contains a symbol instance that you want to receive the copied tween. Uncomment the line that calls the `addTarget()` function and replace the `<instance name goes here>` text in that line with the name of the symbol instance you want to animate.

To name the symbol instance you want to animate with the pasted ActionScript, select the instance on the Stage and enter the name in the Property inspector.

You can also name a motion tween instance by selecting the tween span in the Timeline and entering a name for the motion tween in the Property inspector. You can then reference the tween span in ActionScript 3.0 code.

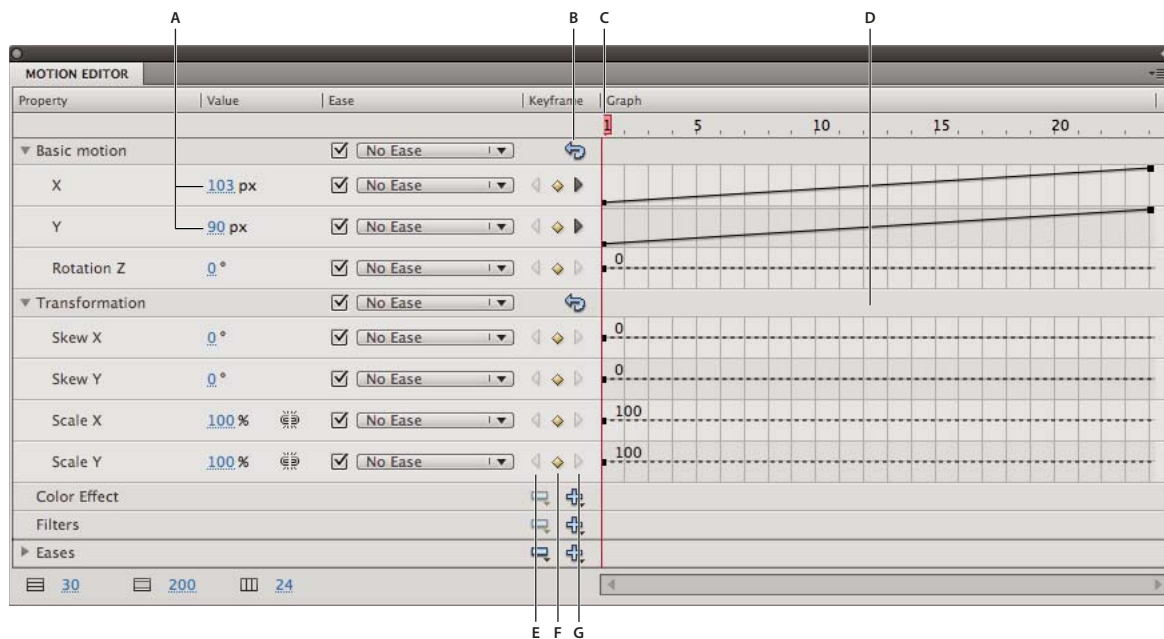
For more information about animating with ActionScript 3.0, see the `fl.motion` classes in the *ActionScript 3.0 Language and Components Reference*.

Editing property curves with the Motion Editor

The Motion Editor panel lets you view all tween properties and their property keyframes. It also provides tools for adding precision and detail to tweens. The Motion Editor displays the properties of the currently selected tween. Once you have created a tween in the Timeline, the Motion Editor allows you to control the tween in several different ways.

The Motion Editor panel allows you to:

- Set the value of individual property keyframes.
- Add or remove property keyframes for individual properties.
- Move property keyframes to different frames within the tween.
- Copy and paste a property curve from one property to another.
- Reverse the keyframes for individual properties.
- Reset individual properties or property categories.
- Exert fine control over the shape of the tween curves of most individual properties using Bezier controls. (The X, Y, and Z properties do not have Bezier controls.)
- Add or remove filters or color effects and adjust their settings.
- Add different preset eases to individual properties and property categories.
- Create custom ease curves.
- Add custom easing to individual tweened properties and groups of properties.
- Enable roving for individual property keyframes for X, Y, and Z properties. Roving allows the property keyframe to move to different frames or between individual frames to create smooth motion.



The Motion Editor.

A. Property values B. Reset Values button C. Playhead D. Property curve area E. Previous Keyframe button F. Add or Remove Keyframe button G. Next Keyframe button

When you select a tween span in the Timeline or a tweened object or motion path on the Stage, the Motion Editor displays the property curves of the tween. The Motion Editor displays the property curves on a grid representing the frames of the Timeline in which the selected tween occurs. The playhead appears in both the Timeline and the Motion Editor, always in the same frame number.

The Motion Editor represents tweened property values using a two-dimensional graph for each property. Each property has its own graph. Each graph represents time horizontally (from left to right), and the change to a property value vertically. Each property keyframe for a specific property appears as a control point on the property curve for that property. If an ease curve is applied to a property curve, a second curve appears as a dashed line in the property curve area. The dashed line curve shows the affect of the ease on the property values.

Some properties cannot be tweened because they can have only one value for the life of an object in the Timeline. An example is the Quality property of the Gradient Bevel filter. These properties can be set in the Motion Editor, but do not have graphs.

You can precisely control the shape of most property curves in the Motion Editor by adding property keyframes and manipulating the curve with standard Bezier controls. For the X, Y, and Z properties, you can add and remove control points on the property curves, but you cannot use Bezier controls. When you change the control points of a property curve, the changes appear on Stage immediately.

The Motion Editor also allows you to apply easing to any property curve. Applying easing in the Motion Editor lets you create certain types of complex animation effects without creating complex motion paths. Ease curves are graphs that show how the values of a tweened property are interpolated over time. By applying an ease curve to a property curve, you can create complex motion with little effort.

Some properties have minimum or maximum values that cannot be exceeded, such as Alpha transparency (0-100%). The graphs for these properties cannot apply values outside the acceptable range.

For a video tutorial about the Motion Editor, see Using the Motion Editor at www.adobe.com/go/lrvid4057_fl.

See also

“[Drawing with the Pen tool](#)” on page 102

“[Easing tweens](#)” on page 207

Control the Motion Editor display

In the Motion Editor, you can control which property curves appear, and the size of the display of each property curve. Property curves that are displayed at a large size are easier to edit.

- To adjust which properties appear in the Motion Editor, click the triangle next to a property category to expand or collapse the category.
- To control the number of frames of a tween shown in the Motion Editor, enter the number of frames you want to display in the Viewable Frames field at the bottom of the Motion Editor. The maximum number of frames is the total number of frames in the selected tween span.
- To toggle a property curve between expanded and collapsed views, click the property name. The expanded view provides a lot more room to edit the property curve. You can adjust the size of the collapsed and expanded views with the Graph Size and Expanded Graph Size fields at the bottom of the Motion Editor.
- To enable or disable tooltips in the graph area, select Show Tooltips from the panel options menu.
- To add a new color effect or filter to a tween, click the Add button in a property category row and choose the item you want to add. The new item appears in the Motion Editor immediately.

Edit the shape of property curves

The Motion Editor allows you to precisely control the shape of each property curve of a tween, except for X, Y, and Z. For all other properties, you can edit the curve of each graph with standard Bezier controls. Using these controls is similar to how you edit strokes with the Selection tool or the Pen tool. Move a curve segment or control point upward to increase the property value, move it downward to decrease the value.

By working with property curves directly, you can:

- Create complex curves for complex tweened effects.
- Adjust property values at property keyframes.
- Increase or decrease property values along the entire property curve.
- Add additional property keyframes to a tween.
- Set individual property keyframes to roving or non-roving.

In the Motion Editor, the basic motion properties X, Y, and Z are different from other properties. These three properties are linked together. A frame in a tween span that is a property keyframe for one of these properties must be a property keyframe for all three of them. In addition, control points on the X, Y, and Z property curves cannot be edited with Bezier controls.

Control points of property curves can be either smooth points or corner points. When a property curve passes through a corner point, it forms an angle. When a property curve passes through a smooth point, it forms a smooth curve. For X, Y, and Z, the type of a control point in the property curve depends on the type of the corresponding control point in the motion path on the Stage.

In general it is best to edit the X, Y, and Z properties of a tween by editing the motion path on the Stage. Use the Motion Editor for minor adjustments to the property values or to move their property keyframes to different frames of the tween span.

- To change the shape of a curve segment between two control points, drag the segment. When you drag a curve segment, the control points at each end of the segment become selected. If the selected control points are smooth points, their Bezier handles appear.
- To reset a property curve to a static, non-tweened property value, right-click (Windows) or Command-click (Macintosh) the property graph area and choose Reset Property.
- To reset an entire category of properties to static, non-tweened values, click the Reset Values button for the category.
- To reverse the direction of a property tween, right-click (Windows) or Command-click (Macintosh) the property graph area and choose Reverse Keyframes.
- To copy a property curve from one property to another, right-click (Windows) or Command-click (Macintosh) the graph area of the curve and choose Copy Curve. To paste the curve to another property, the graph area of the property and choose Paste Curve. You can also copy curves between custom eases and between custom eases and properties.

Work with property keyframes

You can edit the shape of a property curve by adding, removing, and editing property keyframes along each graph.

- To add a property keyframe to a property curve, place the playhead in the desired frame and click the Add or Remove Keyframe button for the property in the Motion Editor.

You can also Ctrl-click (Windows) or Option-click (Macintosh) the graph in the frame where you want to add the property keyframe.

You can also right-click (Windows) or Command-click (Macintosh) the property curve and choose Add Keyframe.

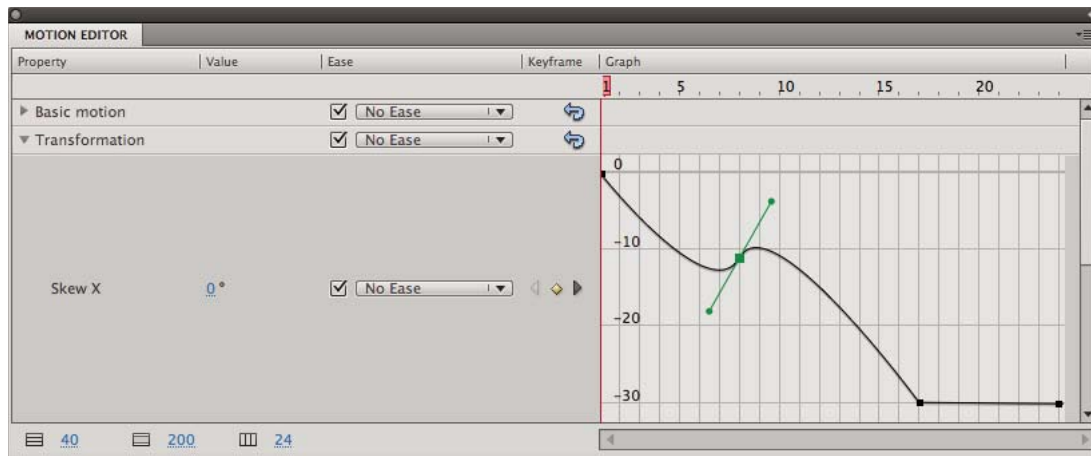
- To remove a property keyframe from a property curve, Ctrl-click (Windows) or Option-click (Macintosh) the control point for the property keyframe in the property curve.

You can also right-click (Windows) or Command-click (Macintosh) the control point and choose Remove Keyframe.

- To toggle a control point between corner point and smooth point modes, Alt-click (Windows) or Command-click (Macintosh) the control point.

When a control point is in smooth point mode, its Bezier handles are exposed and the property curve passes through the point as a smooth curve. When a control point is a corner point, the property curve forms an angle when it passes through the control point. Bezier handles are not exposed for corner points.

To set the point to smooth point mode you can also right-click (Windows) or Command-click (Macintosh) the control point and choose Smooth Point, Smooth Right, or Smooth Left. To set the point to corner point mode, choose Corner Point.



A property curve showing a smooth point (frame 8) and a corner point (frame 17).

- To move a property keyframe to a different frame, drag its control point.
You cannot drag a property keyframe past the keyframes that follow or precede it.
- To toggle a property keyframe for the spatial properties X, Y, and Z between roving and non-roving, right-click (Windows) or Command-click (Macintosh) the property keyframe in the Motion Editor. For more information about roving keyframes, see [“Edit the motion path of a tween”](#) on page 195.

You can also turn off roving for an individual property keyframe in the Motion Editor by dragging the roving keyframe to a vertical frame divider.

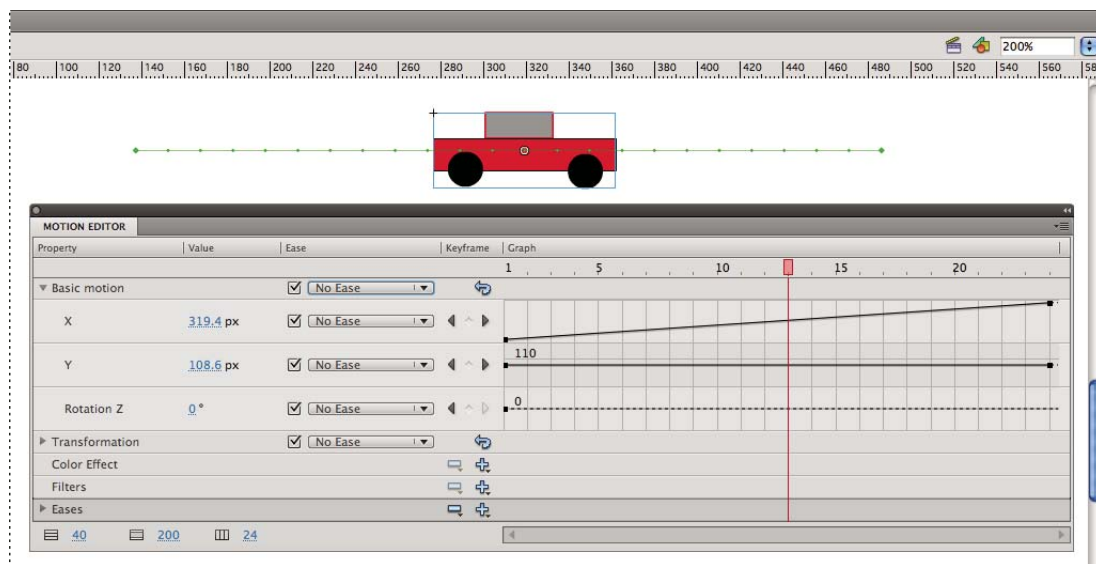
- To link associated pairs of X and Y properties, click the Link X and Y Property Values button for either one of the properties you want to link. When the properties are linked, their values are constrained so the ratio between them is preserved when you enter a value for either of the linked properties. Examples of associated X and Y properties include the Scale X and Scale Y properties and the Blur X and Blur Y properties of the Drop Shadow filter.

Easing tweens

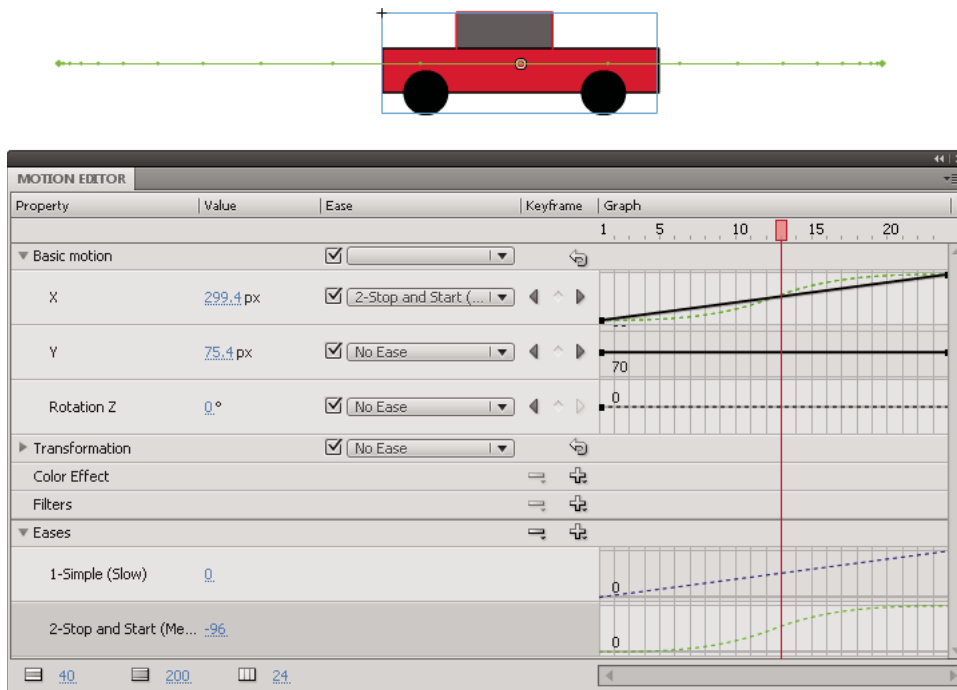
Easing is a technique for modifying the way that Flash calculates the property values in between property keyframes in a tween. Without easing, Flash calculates these values so that the change to the value is the same in each frame. With easing, you can adjust the rate of change to the values for more natural or more complex animation.

An ease is a mathematical curve that is applied to the property values of a tween. The final effect of the tween is the result of the combination of the range of property values in the tween and the ease curve.

For example, if you animate a picture of a car across the Stage, the motion is more realistic if the car starts from a stop and slowly gains speed and then slowly comes to a stop at the other side of the Stage. Without easing, the car would go from a stop to full speed instantly and then stop instantly at the other side of the Stage. With easing, you can apply a motion tween to the car and then make that tween start and end slowly.



A motion path with no ease applied. Note the even distribution of frames along the path.



The same motion path with the Stop and Start (Medium) ease applied. Note the concentration of frames near the ends of the path, resulting in more realistic acceleration and deceleration of the car.

You can apply easing in the Property inspector or the Motion Editor. Eases applied in the Property inspector affect all of the properties that are included in a tween. Eases applied in the Motion Editor can affect a single property, a group of properties, or all properties of a tween.

An ease can be simple or complex. Flash includes a range of preset eases that you can apply for simple or complex effects. In the Motion Editor, you can also create your own custom ease curves.

One of the many common uses of easing is when you have edited a motion path on the Stage and enabled roving keyframes to make the speed of motion consistent in each segment of the path. You can then use easing to add more realistic acceleration and deceleration at the ends of the path.

When you apply an ease curve to a property curve, a visual overlay of the ease curve appears in the property curve graph area. By showing the property curve and the ease curve in the same graph area, the overlay makes it easier to understand the final tween effect that you see on the Stage when testing the animation.

Because ease curves in the Motion Editor can be complex, you can use them to create complex motion on the Stage without creating complex motion paths on the Stage. You can also use ease curves to create complex tweens of any other properties in addition to the spatial properties X, Y, and Z.

Jen DeHaan provides a useful article about [modifying and applying custom eases](#) and another about [easing between keyframes](#) at her [Flashthusiast.com](#) site.

Ease all properties of a tween in the Property inspector

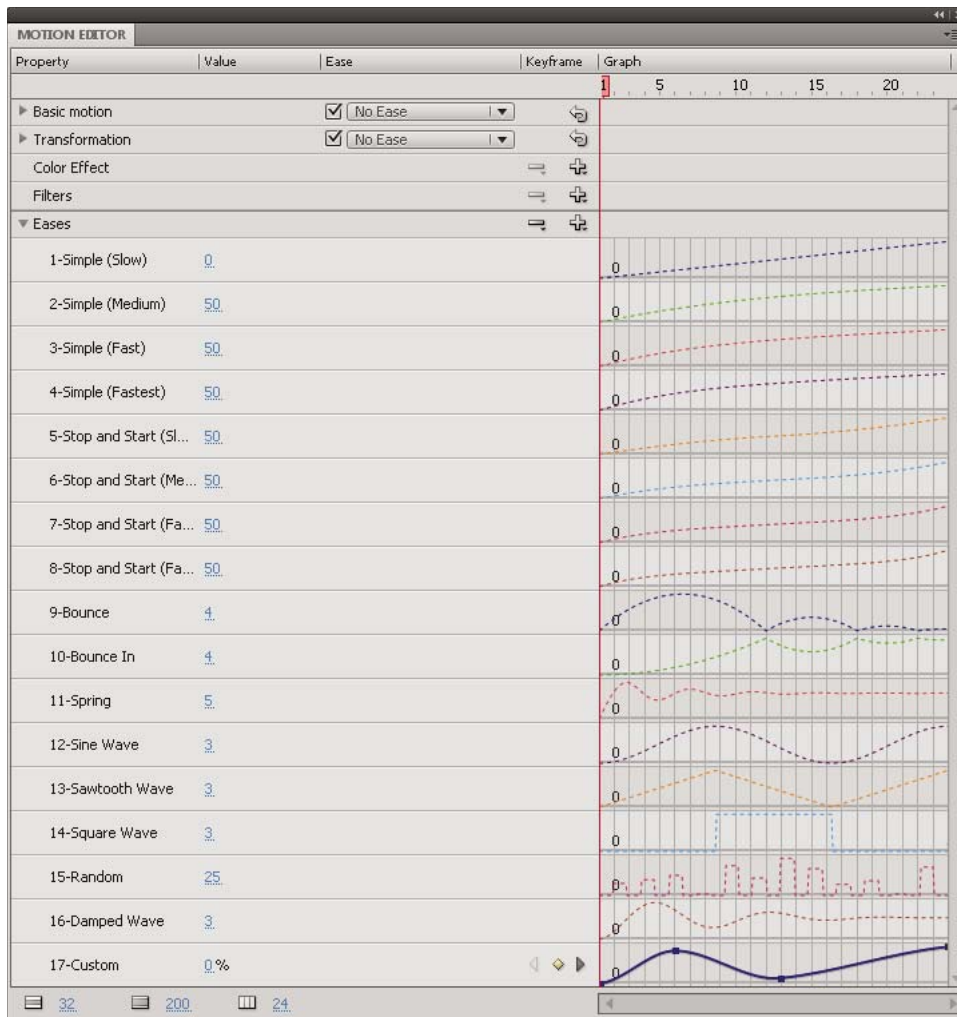
When you apply an ease to a tween with the Property inspector, the ease affects all of the properties that are included in the tween. The Property inspector applies the Simple (Slow) ease curve, which is also available in the Motion Editor.

- 1 Select the tween in the Timeline or the motion path on the Stage.
- 2 In the Property inspector, select the ease to apply from the Ease menu.

3 Enter a value for the strength of the ease in the Ease Value field.

Ease individual properties in the Motion Editor

You can apply a preset ease to an individual property or a category of properties in the Motion Editor.



The shapes of the ease curves included with Flash.

To use eases in the Motion Editor, add an ease curve to the list of eases available to the selected tween and then apply the ease to the properties you choose. When you apply an ease to a property, a dashed-line curve appears overlaid onto the graph area of the property. The dashed-line curve shows the affect of the ease curve on the actual values of that property of the tween.

- To add an ease to the list of eases available to the selected tween, click the Add button in the Ease section of the Motion Editor and then select the ease to add.
- To add an ease to a single property, select the ease from the Selected Ease menu for the property.
- To add an ease to a whole category of properties, such as motion, transformation, color effect, or filters, select an ease type from the Selected Ease menu for the property category.

- To enable or disable the effect of the ease for a property or category of properties, click the Enable/Disable Ease checkbox for the property or category of properties. This allows you to quickly see the effect of an ease on property curves.
- To remove an ease from the list of available tweens, click the Remove Ease button in the Eases section of the Motion Editor and then select the ease from the pop-up menu.

Edit ease curves in the Motion Editor

In the Motion Editor, you can edit the properties of the preset ease curves and make your own custom ease curves.

- To edit a preset ease curve, set the value for the ease in the field next to the ease name.

For simple ease curves, the value is a percentage representing the strength of the application of the ease curve to the property curve. Positive values increase the ease at the end of the curve. Negative values increase the ease at the beginning of the curve.

For ease curves that are waves, such as the sine wave or sawtooth wave, the value represents the number of half-cycles in the wave.

To edit a preset ease, first add it to the list of available eases with the Add button in the Ease section of the Motion Editor.

- To edit a custom ease curve, add an instance of the Custom ease curve to the list of eases and then edit the curve using the same techniques as for editing any other Bezier curve in Flash. The initial value of the ease curve must always be 0%.

You can also copy and paste the curve from one custom ease to another, including to custom eases in separate motion tweens.

Working with classic tweens

Classic tweens are similar to motion tweens, except that they are more complex to create and enable certain types of animated effects not possible with motion tweens.

About classic tweens

Classic tween animation in Flash is similar to motion tweens, but is somewhat more complicated to create and less flexible. However, classic tweens provide some types of control over animation that motion tweens do not. For more information about the differences, see “[Differences between motion tweens and classic tweens](#)” on page 189.

For samples of classic tween animation, see the Flash Samples page at www.adobe.com/go/learn_fl_samples. The following samples are available:

- Animated Drop Shadow: Download and decompress the Samples ZIP file and navigate to the Graphics\AnimatedDropShadow folder to access the sample.
- Animation and Gradients: Download and decompress the Samples ZIP file and navigate to the Graphics\AnimationAndGradients folder to access the sample.

See also

“[About tweened animation](#)” on page 187

Create and edit keyframes for classic tweens

Changes in the animation are defined in a *keyframe*. When you create frame-by-frame animation, every frame is a keyframe. In tweened animation, you define keyframes at significant points in the animation and Flash creates the contents of frames between. The interpolated frames of a tweened animation appear as light blue or light green with an arrow drawn between keyframes. Because Flash documents save the shapes in each keyframe, create keyframes only at those points in the artwork where something changes.

Keyframes are indicated in the Timeline: a solid circle represents a keyframe with content on it, and an empty circle before the frame represents an empty keyframe. Subsequent frames added to the same layer have the same content as the keyframe.

Only keyframes are editable in a classic tween. You can view tweened frames, but you can't edit them directly. To edit tweened frames, change one of the defining keyframes or insert a new keyframe between the beginning and ending keyframes. Drag items from the Library panel onto the Stage to add the items to the current keyframe.

To display and edit more than one frame at a time, use onion skinning.

Create keyframes

❖ Do one of the following:

- Select a frame in the Timeline, and select Insert > Timeline > Keyframe.
- Right-click (Windows) or Control-click (Macintosh) a frame in the Timeline and select Insert Keyframe.

Insert frames in the Timeline

- To insert a new frame, select Insert > Timeline > Frame.
- To create a new keyframe, select Insert > Timeline > Keyframe, or right-click (Windows) or Control-click (Macintosh) the frame where you want to place a keyframe, and select Insert Keyframe.
- To create a new blank keyframe, select Insert > Timeline > Blank Keyframe, or right-click (Windows) or Control-click (Macintosh) the frame where you want to place the keyframe, and select Insert Blank Keyframe.

Delete or modify a frame or keyframe

- To delete a frame, keyframe, or frame sequence, select it and right-click (Windows) or Control-click (Macintosh) and select Remove Frames. Surrounding frames remain unchanged.
- To move a keyframe or frame sequence and its contents, select it and drag to the desired location.
- To extend the duration of a keyframe, Alt-drag (Windows) or Option-drag (Macintosh) it to the final frame of the new sequence.
- To copy and paste a frame or frame sequence, select it and select Edit > Timeline > Copy Frames. Select a frame or sequence to replace, and select Edit > Timeline > Paste Frames.
- To convert a keyframe to a frame, select the keyframe and select Modify > Timeline > Clear Keyframe, or right-click (Windows) or Control-click (Macintosh) it and select Clear Keyframe. The cleared keyframe and all frames up to the subsequent keyframe are replaced with the contents of the frame preceding the cleared keyframe.
- To copy a keyframe or frame sequence by dragging, select it and Alt-drag (Windows) or Option-drag (Macintosh) to the new location.
- To change the length of a tweened sequence, drag the beginning or ending keyframe left or right.
- To add a library item to the current keyframe, drag the item from the Library panel onto the Stage.
- To reverse an animation sequence, select the appropriate frames in one or more layers and select Modify > Timeline > Reverse Frames. Keyframes must be at the beginning and end of the sequence.

Add a classic tween to instances, groups, or type

To tween the changes in properties of instances, groups, and type, you can use classic tweens. Flash can tween position, size, rotation, and skew of instances, groups, and type. Additionally, Flash can tween the color of instances and type, creating gradual color shifts or making an instance fade in or out.

Before tweening the color of groups or type, make them into symbols. Before animating individual characters in a block of text separately, place each character in a separate text block.

If you apply a classic tween and then change the number of frames between the two keyframes, or move the group or symbol in either keyframe, Flash automatically tweens the frames again.

For a video tutorial about classic tweens, see www.adobe.com/go/vid0125.

For a text tutorial about classic tweens, see Create a Timeline Animation on the Flash Tutorials page at www.adobe.com/go/learn_fl_tutorials.

For samples of classic tween animation, see the Flash Samples page at www.adobe.com/go/learn_fl_samples. The following samples are available:

- Animated Drop Shadow: Download and decompress the Samples zip file and navigate to the Graphics\AnimatedDropShadow folder to access the sample.
- Animation and Gradients: Download and decompress the Samples zip file and navigate to the Graphics\AnimationAndGradients folder to access the sample.

See also

“[Create symbols](#)” on page 148

“[Break text apart](#)” on page 252

“[About tweened animation](#)” on page 187

Create a classic tween

1 Click a layer name to make it the active layer, and select an empty keyframe in the layer where you want the animation to start. This will be the first frame of the classic tween.

2 To add content to the first frame of the classic tween, do one of the following:

- Create a graphic object with the Pen, Oval, Rectangle, Pencil, or Brush tool, and then convert it to a symbol.
- Create an instance, group, or text block on the Stage.
- Drag an instance of a symbol from the Library panel.

Note: To create a tween, you must have only one item on the layer.

3 Create a second keyframe where you want the animation to end, and leave the new keyframe selected.

4 To modify the item in the ending frame, do any of the following:

- Move the item to a new position.
- Modify the item’s size, rotation, or skew.
- Modify the item’s color (instance or text block only). To tween the color of elements other than instances or text blocks, use shape tweening.

5 To create the classic tween, do one of the following:

- Click any frame in the tween’s frame span and select Insert > Classic Tween.

- Right-click (Windows) or Control-click (Macintosh) any frame in the tween's frame span and select Create Classic Tween from the context menu.

If you created a graphic object in step 2, Flash automatically converts the object to a symbol and names it *tween1*.

- 6 If you modified the size of the item in step 4, select Scale in the Tweening section of the Property inspector to tween the size of the selected item.
- 7 To produce a more realistic sense of motion, apply easing to the classic tween. To apply easing to a classic tween, use the Ease field in the Tweening section of the Property inspector to specify an easing value for each classic tween you create. Use the Custom Ease In/Ease Out dialog box to more precisely control the speed of the classic tween.

Drag the value in the Easing field or enter a value to adjust the rate of change between tweened frames:

- To begin the classic tween slowly and accelerate the tween toward the end of the animation, enter a negative value between -1 and -100.
- To begin the classic tween rapidly and decelerate the tween toward the end of the animation, enter a positive value between 1 and 100.
- To produce a more complex change in speed within the tween's frame span, click the Edit button next to the Ease field to open the Custom Ease In/Ease Out dialog box.

By default, the rate of change between tweened frames is constant. Easing creates a more natural appearance of acceleration or deceleration by gradually adjusting the rate of change.

- 8 To rotate the selected item during the tween, select an option from the Rotate menu in the Property inspector:
 - To prevent rotation, select None (the default setting).
 - To rotate the object once in the direction requiring the least motion, select Auto.
 - To rotate the object as indicated, and then enter a number to specify the number of rotations, select Clockwise (CW) or Counterclockwise (CCW).

Note: The rotation in step 8 is in addition to any rotation you applied to the ending frame in step 4.

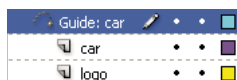
- 9 If you're using a motion path, select Orient To Path in the Property inspector to orient the baseline of the tweened element to the motion path.
- 10 To synchronize the animation of graphic symbol instances with the main Timeline, select the Sync option in the Property inspector.

Note: Modify > Timeline > Synchronize Symbols and the Sync option both recalculate the number of frames in a tween to match the number of frames allotted to it in the Timeline. Use the Sync option if the number of frames in the animation sequence inside the symbol is not an even multiple of the number of frames the graphic instance occupies in the document.

- 11 If you're using a motion path, select Snap to attach the tweened element to the motion path by its registration point.

Create classic tween motion along a path

Motion guide layers let you draw paths along which tweened instances, groups, or text blocks can be animated. You can link multiple layers to a motion guide layer to have multiple objects follow the same path. A normal layer that is linked to a motion guide layer becomes a guided layer.



In this example, two objects on separate layers are attached to the same motion path.

For a video tutorial about classic tweens, see www.adobe.com/go/vid0125.

For a written tutorial about classic tweens, see Create a Timeline Animation on the Flash Tutorials web page at www.adobe.com/go/learn_fl_tutorials.

For samples of classic tween animation, see the Flash Samples page at www.adobe.com/go/learn_fl_samples. The following samples are available:

- Animated Drop Shadow: Download and decompress the Samples zip file and navigate to the Graphics\AnimatedDropShadow folder to access the sample.
- Animation and Gradients: Download and decompress the Samples zip file and navigate to the Graphics\AnimationAndGradients folder to access the sample.

Create a motion path for classic tweened animation

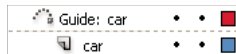
1 Create a classic-tweened animation sequence.

If you select Orient To Path in the Property inspector, the baseline of the tweened element orients to the motion path. If you select Snap, the registration point of the tweened element snaps to the motion path.

2 Right-click (Windows) or Control-click (Macintosh) the layer name of the layer containing the classic tween and choose Add Classic Motion Guide.

Flash adds a motion guide layer above the classic tween layer and indents the name of the classic tween layer to show that it is bound to the motion guide layer.

Note: If you already have a guide layer in the Timeline, you can drag a layer containing the classic tween below the guide layer to convert the guide layer to a motion guide and bind the classic tween to it.

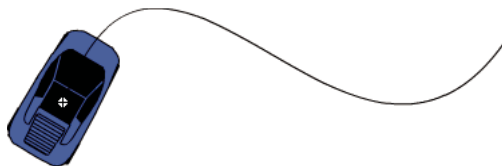


A motion guide layer above the layer containing the classic tween.

3 To add a path to the motion guide layer to guide the classic tween, select the motion guide layer and use the Pen, Pencil, Line, Circle, Rectangle, or Brush tool to draw the desired path.

You can also paste a stroke onto the motion guide layer.

4 Drag the object you are tweening to snap it to the beginning of the line in the first frame, and to the end of the line in the last frame.



A graphic of a car snapped to the beginning of a guide stroke.

Note: For best snapping results, drag the symbol by its transformation point.

5 To hide the motion guide layer and the path so that only the object's movement is visible while you work, click in the Eye column on the motion guide layer.

The group or symbol follows the motion path when you play the animation.

Link layers to a motion guide layer

- ❖ Do one of the following:
 - Drag an existing layer below the motion guide layer. The layer is indented under the motion guide layer. All objects on this layer automatically snap to the motion path.
 - Create a new layer under the motion guide layer. Objects you tween on this layer are automatically tweened along the motion path.
 - Select a layer below a motion guide layer. Select Modify > Timeline > Layer Properties, and select Guide.

Unlink layers from a motion guide layer

- ❖ Select the layer to unlink and do one of the following:
 - Drag the layer above the motion guide layer.
 - Select Modify > Timeline > Layer Properties, and select Normal as the layer type.

Paste classic tween properties

The Paste Motion command lets you copy a classic tween, and paste only specific properties to apply to another object.

- 1 Select the frames in the Timeline that contain the classic tween to copy. The frames you select must be on the same layer, however, they do not have to span a single classic tween. The selection can span a tween, empty frames, or two or more tweens.
- 2 Select Edit > Timeline > Copy Motion.
- 3 Select the symbol instance to receive the copied classic tween.
- 4 Select Edit > Timeline > Paste Motion Special. Select the specific classic tween properties to paste to the symbol instance. The classic tween properties are:

X Position How far an object moves in the x direction.

Y Position How far an object moves in the y direction.

Horizontal Scale The ratio between the current size of the object and its natural size in the horizontal direction (X).

Vertical Scale Specifies the ratio between the current size of the object and its natural size in the vertical direction (Y).

Rotation And Skew The rotation and skew of the object. These properties must be jointly applied to an object. Skew is a measurement of rotation in degrees, and when you rotate and skew, each property affects the other.

Color All color values such as Tint, Brightness, and Alpha are applied to the object.

Filters All filter values and changes for the selected span. If filters are applied to an object, the filter is pasted with all values intact, and its state (enabled or disabled) applies to the new object as well.

Blend Mode Applies the blend mode of the object.

Override Target Scale Properties When unchecked, specifies that all properties be pasted relative to the target object. When checked, this option overrides the scale properties of the target.

Override Target Rotation And Skew Properties When unchecked, specifies that all properties be pasted relative to the target object. When checked, the pasted properties override the existing rotation and scale properties of the object.

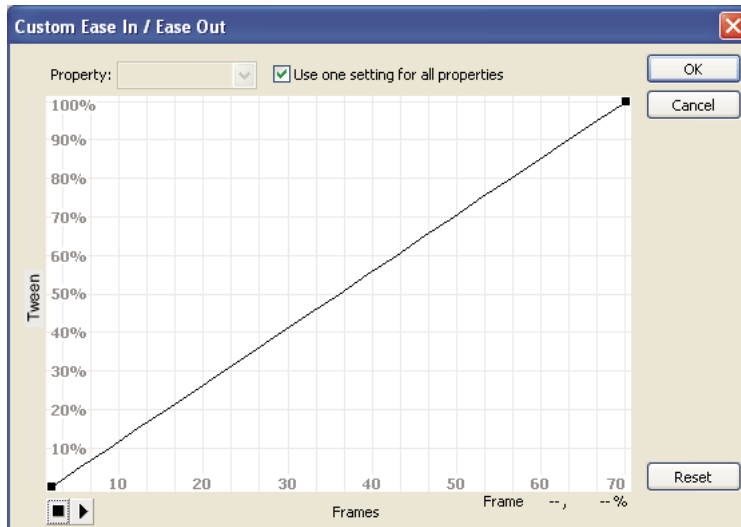
The necessary frames, tween, and symbol information are inserted to match the original, copied tween.

To copy a symbol's classic tween to the Actions panel or use it in another project as ActionScript, use the Copy Motion as ActionScript 3.0 command.

Apply custom ease in/ease out to classic tweens

The Custom Ease In/Ease Out dialog box displays a graph representing the degree of motion over time. The horizontal axis represents frames, and the vertical axis represents percentage of change. The first keyframe is represented as 0%, and the last keyframe is represented as 100%.

The slope of the graph's curve represents the rate of change of the object. When the curve is horizontal (no slope), the velocity is zero; when the curve is vertical, an instantaneous rate of change occurs.



Custom Ease In/Ease Out graph showing constant velocity

Additional controls for the Custom Ease In/Ease Out dialog box

Use One Setting For All Properties check box The default is selected; the displayed curve is used for all properties, and the Properties pop-up menu is disabled. When the check box is not selected, the Properties pop-up menu is enabled, and each property has a separate curve defining the velocity of that property.

Property pop-up menu Enabled only when the Use One Setting for All Properties check box is not selected. When enabled, a separate curve is maintained for each of the five properties appearing in the menu. Selecting a property in the menu displays the curve for that property. The properties are:

- **Position** Specifies custom ease settings for the position of an animated object on the Stage.
- **Rotation** Specifies custom ease settings for the rotation of an animated object. For example, you can fine-tune how quickly or slowly an animated character turns around to face the user on the Stage.
- **Scale** Specifies custom ease settings for the scale of an animated object. For example, you can more easily customize the scale of an object so it appears to be moving away from the viewer, then coming closer, and then moving away again.
- **Color** Specifies custom ease settings for color transitions applied to an animated object.
- **Filters** Specifies custom ease settings for filters applied to an animated object. For example, you can control the ease setting of a drop shadow that simulates a change in the direction of a light source.


Play and Stop buttons Let you preview an animation on the Stage using all the current velocity curves defined in the Custom Ease In/Ease Out dialog box.

Reset button Lets you reset the velocity curve to the default, linear state.

Position of the selected control point In the lower-right corner of the dialog box, a numeric value displays the keyframe and position of the selected control point. If no control point is selected, no value appears.

To add a control point to the line, click the diagonal line once. To achieve a precise degree of control over the motion of an object, drag the positions of the control points.

Using frame indicators (represented by square handles), click where you want an object to slow down or speed up. Clicking the square handle of a control point selects that control point, and displays the tangent points on either side of it. Hollow circles represent tangent points. Drag the control point or its tangent points with the mouse or use the keyboard's arrow keys to position them.

 *By default, the control points snap to a grid. You can turn off snapping by pressing the X key while dragging the control point.*

Clicking an area of the curve away from any control points adds a new control point to the curve at that point, without changing the shape of the curve. Clicking away from the curve and control points deselects the currently selected control point.

Add a custom ease

- 1 Select a layer in the Timeline that has a classic tween applied to it.
- 2 Click the Edit button next to the Ease slider in the frame Property inspector.
- 3 (Optional) To display the curve for a property, deselect Use One Setting For All Properties, and select a property in the menu.
- 4 To add a control point, Control-click (Windows) or Command-click (Macintosh) the diagonal line.
- 5 To increase the speed of the object, drag the control point up; to slow down the speed of the object, drag it downwards.
- 6 To further adjust the ease curve, and fine tune the ease value of the tween, drag the vertex handles.
- 7 To view the animation on the Stage, click the play button in the lower-left corner.
- 8 Adjust the controls until you achieve the desired effect.

Note: *If you use the Custom Ease In/Ease Out dialog box to apply a custom ease to a frame, the edit box that shows the ease value shows '--'. If you use the Edit box or the pop-up slider to apply an ease value to a frame, the Custom Ease graph is set to the equivalent curve, and the Use One Setting For All Properties check box is selected.*

Copy and paste an ease curve

- To copy the current ease curve, press Control+C (Windows) or Command+C (Macintosh).
- To paste the copied curve into another ease curve, press Control+V (Windows) or Command+V (Macintosh).

You can copy and paste the ease curve. The copied curve remains available until you exit the Flash application.

Unsupported easing curves

Certain types of easing curves are not supported. No part of the graph can represent a nonlinear curve (such as a circle).

The Custom Ease dialog box automatically prevents moving a control point or a tangent handle to a position that would render an invalid curve:

- All points must exist on the graph. Control points cannot be moved beyond the bounds of the graph.
- All segments of the curve must exist within the graph. The shape of the curve is flattened to prevent it from extending beyond the bounds of the graph.

Shape tweening

About shape tweens

In shape tweening, you draw a vector shape at one specific frame in the Timeline, and change that shape or draw another shape at another specific frame. Flash then interpolates the intermediate shapes for the frames in between, creating the animation of one shape morphing into another.

Shape tweens work best with simple shapes. Avoid shapes with cutouts or negative spaces in them. Experiment with the shapes you want to use to determine the results. You can use shape hints to tell Flash which points on the beginning shape should correspond to specific points on the end shape.

You can also tween the position and color of shapes within a shape tween.

To apply shape tweening to groups, instances, or bitmap images, break these elements apart.

To apply shape tweening to text, break the text apart twice to convert the text to objects.

Create a shape tween

The following steps show how to create a shape tween from frame 1 to frame 30 of the Timeline. However, you can create tweens in any part of the Timeline that you choose.

- 1 In frame 1, draw a square with the Rectangle tool.
- 2 Select frame 30 of the same layer and add a blank keyframe by choosing Insert > Timeline > Blank Keyframe or pressing F7.

- 3 On the Stage, draw a circle with the Oval tool in frame 30.

You should now have a keyframe in frame 1 with a square and a keyframe in frame 30 with a circle.

- 4 In the Timeline, select one of the frames in between the two keyframes in the layer containing the two shapes.
- 5 Choose Insert > Shape Tween.

Flash interpolates the shapes in all the frames between the two keyframes.

- 6 To preview the tween, scrub the playhead across the frames in the Timeline, or press the Enter key.
- 7 To tween motion in addition to shape, move the shape in frame 30 to a location on the Stage that is different from the location of the shape in frame 1.

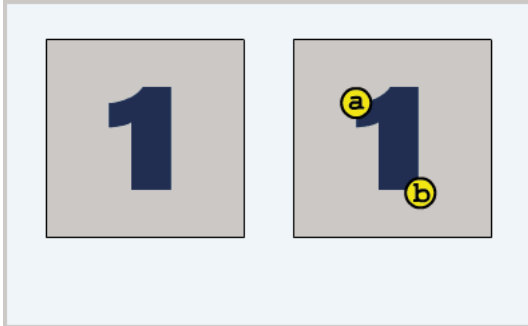
Preview the animation by pressing the Enter key.

- 8 To tween the color of the shape, make the shape in frame 1 a different color from the shape in frame 30.
- 9 To add easing to the tween, select one of the frames between the two keyframes and enter a value in the Ease field in the Property inspector.

Enter a negative value to ease the beginning of the tween. Enter a positive value to ease the end of the tween.

Control shape changes with shape hints

To control more complex or improbable shape changes, you can use shape hints. Shape hints identify points that should correspond in starting and ending shapes. For example, if you are tweening a drawing of a face as it changes expression, you can use a shape hint to mark each eye. Then, instead of the face becoming an amorphous tangle while the shape change takes place, each eye remains recognizable and changes separately during the shift.



Shape hints contain letters (a through z) for identifying which points correspond in the starting and ending shapes. You can use up to 26 shape hints.

Shape hints are yellow in a starting keyframe, green in an ending keyframe, and red when not on a curve.

For best results when tweening shapes, follow these guidelines:

- In complex shape tweening, create intermediate shapes and tween them instead of just defining a starting and ending shape.
- Make sure that shape hints are logical. For example, if you're using three shape hints for a triangle, they must be in the same order on the original triangle and on the triangle to be tweened. The order cannot be *abc* in the first keyframe and *acb* in the second.
- Shape hints work best if you place them in counterclockwise order beginning at the top-left corner of the shape.

For a video tutorial about shape tweens, see www.adobe.com/go/vid0124.

Use shape hints

- 1 Select the first keyframe in a shape-tweened sequence.
- 2 Select **Modify > Shape > Add Shape Hint**. The beginning shape hint appears as a red circle with the letter *a* somewhere on the shape.
- 3 Move the shape hint to a point to mark.
- 4 Select the last keyframe in the tweening sequence. The ending shape hint appears somewhere on the shape as a green circle with the letter *a*.
- 5 Move the shape hint to the point in the ending shape that should correspond to the first point you marked.
- 6 To view how the shape hints change the shape tweening, play the animation again. To fine-tune the tweening, move the shape hints.
- 7 Repeat this process to add additional shape hints. New hints appear with the letters that follow (*b*, *c*, and so on).

View all shape hints

- ❖ Select **View > Show Shape Hints**. The layer and keyframe that contain shape hints must be active for Show Shape Hints to be available.

Remove a shape hint

- ❖ Drag it off the Stage.

Remove all shape hints

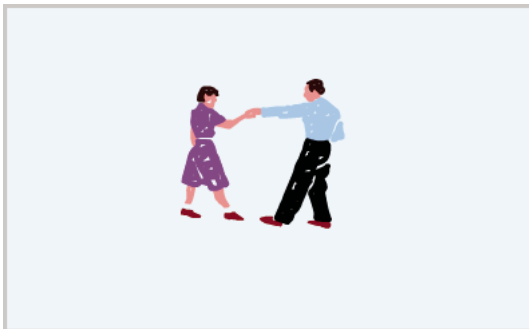
- ❖ Select Modify > Shape > Remove All Hints.

Frame-by-frame animation

Create frame-by-frame animations

Frame-by-frame animation changes the contents of the Stage in every frame and is best suited to complex animation in which an image changes in every frame instead of simply moving across the Stage. Frame-by-frame animation increases file size more rapidly than tweened animation. In frame-by-frame animation, Flash stores the values for each complete frame.

To create a frame-by-frame animation, define each frame as a keyframe and create a different image for each frame. Each new keyframe initially contains the same contents as the keyframe preceding it, so you can modify the frames in the animation incrementally.




- 1 Click a layer name to make it the active layer, and select a frame in the layer where the animation is to start.
- 2 If the frame isn't already a keyframe, select Insert > Timeline > Keyframe.
- 3 Create the artwork for the first frame of the sequence. Use the drawing tools, paste graphics from the Clipboard, or import a file.
- 4 To add a new keyframe whose contents are the same as those of the first keyframe, click the next frame to the right in the same row and select Insert > Timeline > Keyframe, or right-click (Windows) or Control-click (Macintosh) and select Insert Keyframe.
- 5 To develop the next increment of the animation, alter the contents of this frame on the Stage.
- 6 To complete your frame-by-frame animation sequence, repeat steps 4 and 5 until you've built the desired motion.
- 7 To test the animation sequence, select Control > Play or click the Play button on the Controller (Window > Toolbars > Controller).



Use onion skinning

Usually, one frame of the animation sequence at a time appears on the Stage. To help position and edit a frame-by-frame animation, view two or more frames on the Stage at once. The frame under the playhead appears in full color, while surrounding frames are dimmed, making it appear as if each frame were drawn on a sheet of translucent onion-skin paper and the sheets were stacked on top of each other. Dimmed frames cannot be edited.

Simultaneously view several frames of an animation on the Stage

- ❖ Click the Onion Skin button . All frames between the Start Onion Skin and End Onion Skin markers (in the Timeline header) are superimposed as one frame in the Document window.

Control onion skinning display

- To display onion skinned frames as outlines, click the Onion Skin Outlines button .
- To change the position of either onion skin marker, drag its pointer to a new location. (Normally, the onion skin markers move in conjunction with the current frame pointer.)
- To enable editing of all frames between onion skin markers, click the Edit Multiple Frames button . Usually, onion skinning lets you edit only the current frame. However, you can display the contents of each frame between the onion skin markers, and make each available for editing, regardless of which is the current frame.

Note: Locked layers (those with a padlock icon) aren't displayed when onion skinning is turned on. To avoid a multitude of confusing images, lock or hide the layers you don't want to be onion skinned.

Change the display of onion skin markers

- ❖ Click the Modify Onion Markers  button and select an item:

Always Show Markers Displays the onion skin markers in the Timeline header whether or not onion skinning is on.

Anchor Onion Locks the onion skin markers to their current position in the Timeline header. Usually, the onion skin range is relative to the current frame pointer and the onion skin markers. Anchoring the onion skin markers prevents them from moving with the current frame pointer.

Onion 2 Displays two frames on either side of the current frame.

Onion 5 Displays five frames on either side of the current frame.

Onion All Displays all frames on either side of the current frame.

Using mask layers

About mask layers

For spotlight effects and transitions, use a mask layer to create a hole through which underlying layers are visible. A mask item can be a filled shape, a type object, an instance of a graphic symbol, or a movie clip. Group multiple layers under a single mask layer to create sophisticated effects.

To create dynamic effects, animate a mask layer. For a filled shape used as a mask, use shape tweening; for a type object, graphic instance, or movie clip, use motion tweening. When using a movie clip instance as a mask, animate the mask along a motion path.

To create a mask layer, place a mask item on the layer to use as a mask. Instead of having a fill or stroke, the mask item acts as a window that reveals the area of linked layers beneath it. The rest of the mask layer conceals everything except what shows through the mask item. A mask layer can contain only one mask item. A mask layer cannot be inside a button, and you cannot apply a mask to another mask.

To create a mask layer from a movie clip, use ActionScript. A mask layer created with ActionScript can be applied only to another movie clip. See Using Movie Clips as Masks in *Learning ActionScript 2.0 in Adobe Flash*.

Note: The 3D tools cannot be used on objects on mask layers and layers containing 3D objects cannot be used as mask layers. For more information about the 3D tools, see “3D graphics” on page 139.

For a video tutorial about animating masks, see www.adobe.com/go/vid0127.

For a sample of scriptable masks, see the Flash Samples web page at www.adobe.com/go/learn_fl_samples. Download and decompress the Samples zip file and navigate to the Masking\ScriptableMasks folder to access the sample.

Work with mask layers

You can use mask layers to reveal portions of a picture or graphic in the layer below. To create a mask, you specify that a layer is a mask layer, and either draw or place a filled shape on that layer. You can use any filled shape, including groups, text, and symbols, as a mask. The mask layer reveals the area of linked layers beneath the filled shape.

For a video tutorial about animating masks, see www.adobe.com/go/vid0127.

For a sample of scriptable masks, see the Flash Samples web page at www.adobe.com/go/learn_fl_samples. Download and decompress the Samples zip file and navigate to the Masking\ScriptableMasks folder to access the sample.

See also

“Motion tweens” on page 187

Create a mask layer

- 1 Select or create a layer containing the objects to appear inside the mask.
- 2 Select Insert > Timeline > Layer to create a new layer above it. A mask layer always masks the layer immediately below it; create the mask layer in the proper place.
- 3 Place a filled shape, text, or an instance of a symbol on the mask layer. Flash ignores bitmaps, gradients, transparency, colors, and line styles in a mask layer. Any filled area is completely transparent in the mask; any non-filled area is opaque.
- 4 Right-click (Windows) or Control-click (Macintosh) the mask layer’s name in the Timeline, and select Mask. A mask layer icon indicates the mask layer. The layer immediately below it is linked to the mask layer, and its contents show through the filled area on the mask. The masked layer name is indented, and its icon changes to a masked layer icon.
- 5 To display the mask effect in Flash, lock the mask layer and the masked layer.

Mask additional layers after creating a mask layer

- ❖ Do one of the following:
 - Drag an existing layer directly below the mask layer.
 - Create a new layer anywhere below the mask layer.
 - Select Modify > Timeline > Layer Properties, and select Masked.

Unlink layers from a mask layer

- ❖ Select the layer to unlink and do one of the following:
 - Drag the layer above the mask layer.
 - Select Modify > Timeline > Layer Properties, and select Normal.

Animate a filled shape, type object, or graphic symbol instance on a mask layer

- 1 Select the mask layer in the Timeline.
- 2 To unlock the mask layer, click in the Lock column.
- 3 Do one of the following:
 - If the mask object is a filled shape, apply shape tweening to the object.
 - If the mask object is a type object or graphic symbol instance, apply motion tweening to the object.
- 4 When the animation operation is complete, click in the Lock column for the mask layer to re-lock the layer.

Animate a movie clip on a mask layer

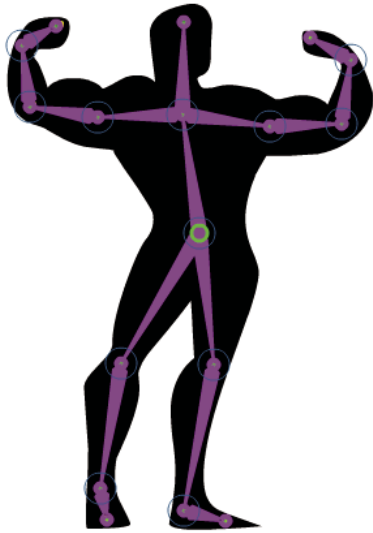
- 1 Select the mask layer in the Timeline.
- 2 To edit the movie clip in place and to display the movie clip's Timeline, double-click the movie clip on the Stage.
- 3 Apply motion tweening to the movie clip.
- 4 When the animation procedure is complete, click the Back button to return to document-editing mode.
- 5 To lock the layer again, click in the Lock column for the mask layer.

Using inverse kinematics

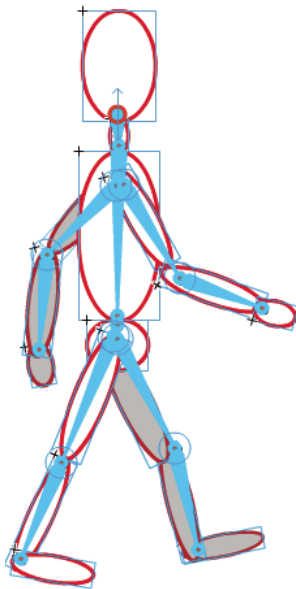
About inverse kinematics

Inverse kinematics (IK) is a method for animating an object or set of objects in relation to each other using an articulated structure of *bones*. Bones allow symbol instances and shape objects to move in complex and naturalistic ways with a minimum of design effort. For example, inverse kinematics lets you create character animation, such as arms, legs, and facial expressions much more easily.

You can add bones to separate symbol instances or to the interior of a single shape. When one bone moves, the other connected bones move in relation to the bone that initiated the movement. When animating using inverse kinematics you need only specify the start and end positions of objects. Inverse kinematics lets you create natural motion much more easily.



A shape with an IK armature added.



A group of several symbols with IK bones attached.

A chain of bones is called an armature. The bones in an armature are connected to each other in a parent-child hierarchy. An armature can be linear or branched. Branches of an armature that originate at the same bone are called siblings. The point where one bone connects to another is called a joint.

You can use IK in Flash in two ways. The first is to articulate a series of symbol instances by adding bones to connect each instance to another instance. The bones allow the chain of symbol instances to move together. For example, you might have a set of movie clips that each represents different parts of a human body. By linking the torso, upper arm, lower arm, and hand together, you can create a realistically moving arm. You can create a branched armature to include both arms, both legs, and the head.

The second way to use IK is to add an armature to the interior of a shape object. The shape can be created in merge drawing mode or object drawing mode. The bones allow you to move and animate parts of the shape without the need to draw different versions of the shape or create a shape tween. For example, you could add bones to a simple drawing of a snake to enable the snake to move and curve realistically.

When you add bones to symbol instances or shapes, Flash moves the instance or shape and the associated armature to a new layer in the Timeline. This new layer is called a *pose layer*. Each pose layer can contain only one armature and its associated instances or shape.

Flash includes two tools for working with IK. You add bones to symbol instances and shapes with the Bone tool. You use the Bind tool to adjust the relationships between individual bones and control points of shape objects.

You can animate armatures and their associated symbols or shapes either in the Timeline or with ActionScript 3.0. You animate in the Timeline by defining different poses for the armature in different frames. Flash interpolates the positions of the armature in the frames in between. For information about animating armatures with ActionScript, see the `fl.motion` classes in the *ActionScript 3.0 Language and Components Reference*.

Note: To use inverse kinematics, your FLA file must specify ActionScript 3.0 as the Script setting in the Flash tab of the Publish Settings dialog box.

For a video tutorial about inverse kinematics, see Using Inverse Kinematics at www.adobe.com/go/lrvid4058_fl.

See also

“Drawing modes and graphic objects” on page 94

Add bones to symbols

You can add IK bones to movie clip, graphic, and button instances. To use text, convert it to a symbol first.

Note: You can also break text apart (*Modify > Break Apart*) into separate shapes and use bones with the individual shapes.

When you add bones to symbol instances, you create a chain of linked instances. This is different from using bones with shapes, where the shape becomes a container for the bones. The linked chain of symbol instances can be a simple linear chain or a branched structure, depending on your needs. A character of a snake would need only a linear chain, while a human figure would need a structure with branches for each limb.

Arrange the symbol instances on the Stage in approximately the configuration that you want them in before adding bones to them. You can edit their relative positions after adding bones, but this step can save time later. The symbol instances can be on different layers before adding bones. Flash moves them to a new layer when bones are added.

- 1 Create symbol instances on the Stage.

Arrange the instances in the approximate spatial configuration you want them in before adding bones.

- 2 Select the Bone tool  from the Tools panel.

You can also press the X key to select the Bone tool.

- 3 With the Bone tool, click the symbol instance that is to be the root or head of the armature. Then drag to a separate symbol instance to link it to the root instance.

While dragging, a bone appears. When you release the mouse, a solid bone is displayed between the two symbol instances. Each bone has a head, the round end, and a tail, the pointed end.

When dragging from one instance to another to create a bone, click the first instance at the specific point where you want to attach the bone to the instance. Release the mouse over the specific point on the second instance where you want the bone to attach. You can also edit these attachment points later. Each symbol instance can have only one attachment point.

The first bone in an armature is the root bone. It appears with a circle around the head of the bone.

By default, Flash moves the transformation point of each symbol instance to the location of the joint formed by each bone connection. For the root bone, the transformation point moves to the head of the bone. For the last bone in a branch, the transformation point moves to the tail of the bone. You can disable the automatic movement of the transformation point in the Drawing tab of Preferences (Edit > Preferences).

- 4 To add another bone, drag from the tail of the first bone to the next symbol instance you want to add to the armature.

The pointer changes when rolled over the head or tail of an existing bone. To make it easier to drag the tail of the new bone to the specific location where you want it, you may want to turn off Snap to Objects (View > Snapping > Snap to Objects).

Link objects with bones in the order of the parent-child relationships you want to create. For example, if you are adding bones to a series of movie clips that represent an arm, draw the first bone from the shoulder to the elbow, and a second bone from the elbow to the wrist, and a third from the wrist to the hand.

As you add bones to instances, Flash moves each instance to a new layer in the Timeline. The new layer is called a *pose layer*. All the bones and symbol instances associated with a given armature reside in the pose layer. Each pose layer can contain only one armature. Flash adds the new pose layer to the Timeline in between existing layers to maintain the previous stacking order of objects on the Stage.

- 5 To create a branched armature, click the head of an existing bone where you want the branch to begin and drag to create the first bone of the new branch.

An armature can have as many branches as necessary.

Note: A branch cannot connect to another branch except at its root.

After creating an IK armature, you can drag the bones or the symbol instances in the armature to reposition the instances. Dragging a bone moves its associated instance without allowing it to rotate relative to its bone. Dragging an instance allows it to move as well as rotate relative to its bone. Dragging an instance in the middle of a branch causes the parent bones to articulate with joint rotation. The child bones move with no joint rotation.

Once you have created an armature and all of its associated symbol instances have moved to a pose layer, you can still add new instances from different layers to the armature. Drag a new bone to the new instance and Flash moves the instance to the pose layer of the armature.

Add bones to shapes

A second way of using IK armatures is with shape objects. With shapes, you can add multiple bones to the interior of a single shape. This is different from symbol instances, where each instance can have only one bone. You can also add bones to shapes created in Object Drawing mode.

You add bones to a single shape or to a group of shapes. In either case, all of the shapes must be selected before adding the first bone. Once bones are added to the selection, Flash converts all of the shapes and bones into an *IK shape object* and moves the object to a new pose layer.


Once a shape is converted to an IK shape, it can no longer merge with other shapes outside the IK shape.

- 1 Create a filled shape or shapes on the Stage.

The shape can contain multiple colors and strokes. Edit the shapes so they are as close to their final form as possible. Once you add bones to a shape, the options for editing the shape become more limited.

2 Select the entire shape on the stage.

If the shape contains multiple color areas or strokes, be sure to select the entire shape. Dragging a selection rectangle around the shape ensures that the entire shape is selected.

3 Select the Bone tool  in the Tools panel.

You can also press the X key to select the Bone tool.

4 With the Bone tool, click inside the shape and drag to another location within the shape.

While dragging, a bone appears. When you release the mouse, a solid bone is displayed between the point where you clicked and the point where you released the mouse. Each bone has a head, the round end, and a tail, the pointed end.

The first bone in an armature is the root bone. It appears with a circle around the head of the bone. When adding the first bone, click in the location within the shape where you want the root of the armature to be. You can also edit the locations of the head and tail of each bone later.

When you add the first bone, Flash converts the shape to an IK shape object and moves it to a new layer in the Timeline. The new layer is called a *pose layer*. All the bones and the IK shape object associated with a given armature reside in the pose layer. Each pose layer can contain only one armature. Flash adds the new pose layer to the Timeline in between existing layers to maintain the previous stacking order of objects on the Stage.

Once the shape becomes an IK shape, you can no longer add new strokes to the shape. You can still add or remove control points from the existing strokes of the shape. The IK shape has its own registration point, transform point, and bounding box.

5 To add another bone, drag from the tail of the first bone to another location in within the shape.

The pointer changes when rolled over the head or tail of an existing bone.

The second bone becomes a child of the root bone. Link areas of the shape with bones in the order of the parent-child relationships you want to create. For example, if you are adding bones to a shape that represents an arm, draw the first bone from the shoulder to the elbow, and a second bone from the elbow to the wrist, and a third from the wrist to the hand.

6 To create a branched armature, click the head of an existing bone where you want the branch to begin and drag to create the first bone of the new branch.

An armature can have as many branches as necessary.

Note: A branch cannot connect to another branch except at its root.

7 To move the armature, select the IK shape object with the Selection tool and then drag any of the bones to move them.

Edit IK armatures and objects

After creating bones, there are many ways of editing them. You can reposition the bones and their associated objects, move a bone within an object, change the length of a bone, delete bones, and edit the objects containing the bones.

IK armatures can only be edited in pose layers that contain only an initial pose in the first frame in which the armature appears in the Timeline. After the armature has been repositioned in later frames of the pose layer, changes to the bone structure cannot be made. To edit the armature, delete any additional poses after the first frame of the armature in the Timeline.

If you are simply repositioning an armature for animation purposes, you can make position changes in any frame of the pose layer. Flash converts the frame to a pose frame.

Select bones and associated objects

- To select an individual bone, click the bone with the Selection tool.
 The bone properties appear in the Property inspector. You can also Shift-click to select multiple bones.
- To move the selection to adjacent bones, click the Parent, Child, or Next/Previous sibling buttons in the Property inspector.
- To select all of the bones in the armature, double-click a bone.
 The properties for all the bones appear in the Property inspector.
- To select an entire armature and display the properties of the armature and its pose layer, click a frame in the pose layer containing the armature.
- To select an IK shape, click the shape.
 The IK shape properties appear in the Property inspector.
- To select a symbol instance connected to a bone, click the instance.
 The instance properties appear in the property inspector.

Reposition bones and associated objects

- To reposition a linear armature, drag any bone in the armature.
 If the armature is connected to symbol instances, you can also drag an instance. This also allows you to rotate the instance relative to its bone.
- To reposition a branch of an armature, drag any bone in the branch.
 All of the bones in the branch move. Bones in other branches of the armature do not move.
- To rotate a bone with its child bones without moving the parent bone, Shift-drag the bone.
- To move an IK shape to a new location on the Stage, select the shape and change its X and Y properties in the Property inspector.

Delete bones

- To delete an individual bone and all of its children, click the bone and press the Delete key.
 You can select multiple bones to delete by Shift-clicking each bone.
- To delete all bones from an IK shape or a symbol armature, select the shape or any symbol instance in the armature and choose Modify > Break Apart.
 IK shapes revert to normal shapes.

Move bones relative to the associated shape or symbol

- To move the location of either end of a bone within an IK shape, drag the end of the bone with the Subselection tool.
- To move the location of a bone joint, head, or tail within a symbol instance, use the Transform panel (Window > Transform) to move the transformation point of the instance.
 The bone moves with the transformation point.
- To move an individual symbol instance without moving any other linked instances, Alt-drag (Windows) or Command-drag (Macintosh) the instance, or drag with the Free Transform tool.
 The bones connected to the instance lengthen or shorten to accommodate the new location of the instance.

Edit an IK shape

You can add, delete, and edit control points of the contours in an IK shape using the Subselection tool.

- To move the position of a bone without changing the IK shape, drag the endpoint of a bone.
- To display the control points of the IK shape boundary, click the stroke of the shape.
- To move a control point, drag the control point.
- To add a new control point, click a part of the stroke without any control points.

You can also use the Add Anchor Point tool in the Tools panel.

- To delete an existing control point, click to select it, and then press the Delete key.


You can also use the Delete Anchor Point tool in the Tools panel.

Bind bones to shape points

Depending on the configuration of an IK shape, you may find that the stroke of the shape does not distort in a pleasing way when moving the armature.

By default, the control points of a shape are connected to the bone that is nearest to them. The Bind tool allows you to edit the connections between individual bones and the shape control points. This lets you control how the stroke distorts when each bone moves for more pleasing results.

You can bind multiple control points to a bone and multiple bones to a control point. Clicking a control point or a bone with the Bind tool displays the connections between the bones and control points. You can then change the connections in a variety of ways.

- To highlight the control points connected to a bone, click the bone with the Bind tool .


The connected points are highlighted in yellow while the selected bone is highlighted in red. Control points connected to only one bone appear as squares. Control points connected to more than one bone appear as triangles.

- To add control points to a selected bone, Shift-click an unhighlighted control point.

You can also Shift-drag to select multiple control points to add to the selected bone.

- To remove control points from the bone, Ctrl-click (Windows) or Option-click (Macintosh) a control point that is highlighted in yellow.

You can also Ctrl-drag (Windows) or Option-drag (Macintosh) to remove multiple control points from the selected bone.

- To highlight the bones connected to a control point, click the control point with the Bind tool .

The connected bones are highlighted in yellow while the selected control point is highlighted in red.

- To add other bones to the selected control point, Shift-click a bone.
- To remove a bone from a selected control point, Ctrl-click (Windows) or Option-click (Macintosh) a bone that is highlighted in yellow.

Adjust IK motion constraints

To create more realistic motion of IK armatures, you can control the freedom of motion of specific bones. For example, two bones that are part of an arm could be constrained so that the elbow cannot bend in the wrong direction.

By default, each IK bone has a fixed length assigned when the bone was created. Bones can rotate around their parent joint and along the x and y axis, but they cannot move in ways that require the length of their parent bone to change.

You can enable, disable, and constrain the rotation of a bone and its motion along the x or y axis. By default, bone rotation is enabled and x and y axis motion are disabled. When x or y axis motion is enabled, a bone can move along the x or y axis to an unlimited degree, and the length of the parent bone changes to accommodate the motion.

You can also limit the speed of motion of a bone to create the effect of weight in a bone.

You set these properties in the Property inspector when one or more bones are selected.

- To enable a selected bone to move along the x or y axis and change the length of its parent bone, select Enable in the Joint: X Translation or Joint: Y Translation section of the Property inspector.

A two-headed arrow appears perpendicular to the bone on the joint to indicate that x axis motion is enabled. A two-headed arrow appears parallel to the bone on the joint to indicate that y-axis motion is enabled. If you enable both x and y-translation for a bone, it is easier to position the bone when rotation is disabled for the bone.

- To limit the amount of motion enabled along the x or y axis, select Constrain in the Joint: X Translation or Joint: Y Translation section of the Property inspector and enter a value for the minimum and maximum distance the bone can travel.
- To disable rotation of the selected bone around the joint, deselect the Enable checkbox in the Joint: Rotation section of the Property inspector.

This checkbox is selected by default.

- To constrain rotation of a bone, enter minimum and maximum degrees of rotation in the Joint: Rotation section of the Property inspector.

The degrees of rotation are relative to the parent bone. An arc appears on top of the bone joint indicating the degrees of freedom of rotation.

- To make a selected bone stationary relative to its parent bone, disable rotation and x and y-axis translation. The bone becomes rigid and follows the motion of its parent.
- To limit the speed of motion of a selected bone, enter a value in the Joint Speed field in the Property inspector. Joint speed gives the bone the effect of weight. The maximum value of 100% is equivalent to unlimited speed.

Animate an armature

You animate IK armatures differently from other objects in Flash. With armatures, you simply add frames to the pose layer and reposition the armature on the Stage to create a keyframe. Keyframes in pose layers are called *poses*. Because IK armatures are typically used for animation purposes, each pose layer acts as a tween layer automatically.

However, IK pose layers are different from tween layers because you cannot tween properties other than bone position in the pose layer. To tween other properties of an IK object such as location, transform, color effects, or filters, enclose the armature and its associated objects in a movie clip or graphic symbol. You can then animate the properties of the symbol using the Insert > Motion Tween command and the Motion Editor panel.

You can also animate IK armatures with ActionScript 3.0 at runtime. If you plan to animate an armature with ActionScript, you cannot animate it in the Timeline. The armature can have only one pose in the pose layer and that pose must be in the first frame in which the armature appears in the pose layer.

See also

[“Motion tweens”](#) on page 187

Animate an armature in the Timeline

IK armatures exist on pose layers in the Timeline. To animate armatures in the Timeline, insert poses by right-clicking a frame in a pose layer and selecting Insert Pose. Use the Selection tool to change the configuration of the armature. Flash automatically interpolates the positions of the bones in the frames between poses.

1 In the Timeline, add frames to the pose layer of the armature to make room for the animation you want to create.

You add frames by right-clicking (Windows) or Option-clicking (Macintosh) a frame in the pose layer to the right of any existing frames and choosing Insert Frame. You can add or delete frames later at any time.

You can also drag the last frame of the pose layer to the right to add frames.

2 To add a pose to a frame in the pose layer, do one of the following:

- Place the playhead in the frame where you want to add the pose and then reposition the armature on the Stage.
- Right-click (Windows) or Option-click (Macintosh) the frame in the pose layer and choose Insert Pose.
- Place the playhead in the frame where you want to add the pose and then press the F6 key.

Flash inserts a pose into the pose layer in the current frame. A diamond-shaped pose marker in the frame indicates the new pose.

3 Add additional poses in separate frames to complete the animation to your satisfaction.

4 If you want to change the length of the animation in the Timeline, drag the last frame of the pose layer to the right or left to add or remove frames. Flash repositions the pose frames in proportion to the change in duration of the layer and reinterpolates the frames in between.

When you are finished, scrub the playhead in the Timeline to preview the animation with the armature positions interpolated in between the pose frames.

You can reposition the armature in the pose frames or add new pose frames at any time.

Convert an armature to a movie clip or graphic symbol for additional tweened effects

To apply tweened effects to IK object properties other than bone position, the object must be enclosed in a movie clip or graphic symbol.

To enclose an IK armature on the Timeline in a movie clip or graphic symbol:

1 Select the IK armature and all of its associated objects.

For an IK shape you can simply click the shape. For linked sets of symbol instances, you can click the pose layer in the Timeline or drag a selection marquee around all of the linked symbols on the Stage.

2 Right-click (Windows) or Ctrl-click (Macintosh) the selection and choose Convert to Symbol from the context menu.

3 In the Convert to Symbol dialog box, enter a name for the symbol and choose either Movie Clip or Graphic from the Type menu. Click OK.

Flash creates a symbol with its own timeline containing the pose layer for the armature.

4 To use the new symbol on the main timeline of your FLA file, drag the symbol from the Library to the Stage.

You can now add motion tween effects to the new symbol instance on the Stage.

You can nest symbols containing IK armatures within as many layers of other nested symbols as necessary to create your desired effect.

Prepare an armature for runtime animation with ActionScript 3.0

IK armatures connected to shapes or movie clip instances can be controlled with ActionScript 3.0. Armatures connected to graphic or button symbol instances cannot be controlled with ActionScript.

Only armatures with a single pose can be controlled with ActionScript. Armatures with more than one pose can only be controlled in the Timeline.

- 1 With the Selection tool, select a frame in a pose layer containing an armature.

The Property inspector displays the armature properties.

- 2 In the Property inspector, choose Runtime from the Type menu.

The hierarchy can now be manipulated with ActionScript 3.0 at runtime.

By default, the armature name in the Property inspector is the same as the pose layer name. Use this name to refer to the armature in ActionScript. You can change the name in the Property inspector.

Add easing to IK animation

When you use poses to add animation to an IK armature, you can adjust the speed of the animation in the frames around each pose. By adjusting the speed, you can create more realistic motion. Controlling the acceleration of motion near pose frames is called *easing*.

For example, when you move your arm, it speeds up and slows down at the beginning and end of the motion. By adding easing to an IK pose layer in the Timeline, you can make the armature speed up or slow down before or after each pose frame.

To add easing to frames in a pose layer:

- 1 Click a frame between two pose frames in the pose layer.

When you apply an ease, it affects the frames between the pose frames to the left and right of the selected frame. If you select a pose frame, the ease affects the frames between the selected pose and the next pose in the layer.

- 2 In the Property inspector, select a type of ease from the Ease menu.

The available eases include four *Simple* and four *Stop and Start* eases. The Simple eases slow the acceleration of the motion in the frames immediately following the previous pose frame *or* the frames immediately before the next pose frame. The Strength property of the ease controls which frames are eased as well as the degree of the ease affect.

The Stop and Start eases slow the motion in the frames immediately after the prior pose frame and the frames immediately before the next pose frame in the layer.

Each of the two types of eases has a Slow, Medium, Fast, and Fastest version. The Slow version has the subtlest effect and the Fastest is the most pronounced effect.

These same ease types are available in the Motion Editor when using motion tweens. You can view the curve of each type of ease in the Motion Editor when a motion tween is selected in the Timeline.

- 3 In the Property inspector, enter a value for the strength of the ease.

The default strength is 0, which is equivalent to no easing. The maximum value is 100, which applies the most significant easing effect to the frames preceding the following pose frame. The minimum value is -100, which applies the most significant easing effect to the frames immediately after the preceding pose frame.

When you are finished, scrub the playhead in the Timeline between the 2 pose frames where you applied the ease to preview the eased motion on the Stage.

See also

[“Easing tweens”](#) on page 207

Chapter 8: Filters and Blends

Adobe® Flash® CS4 Professional special effects include filters, which let you add visual effects to text, buttons, and movie clips; and blend modes, which let you create composite images.

About filters

Filter overview

Adobe® Flash® CS4 Professional filters (graphic effects) let you add interesting visual effects to text, buttons, and movie clips. A feature unique to Flash is that you can animate the filters you apply using motion tweens.

Flash blend modes let you create composite images. *Compositing* is the process of varying the transparency or color interaction of two or more overlapping objects. Blending modes also add a dimension of control to the opacity of objects and images. You can use Flash blending modes to create highlights or shadows that let details from an underlying image show through, or to colorize a desaturated image.

About animated filters

You animate filters in the Timeline. Objects on separate keyframes joined by a tween have the parameters for corresponding filters tweened on intermediate frames. If a filter does not have a matching filter (a filter of the same type) at the opposite end of the tween, a matching filter is added automatically to ensure that the effect occurs at the end of the animation sequence.

To prevent motion tweens from functioning incorrectly if a filter is missing at one end of the tween, or if filters are applied in a different order at each end, Flash does the following:

- If you apply a motion tween to a movie clip with filters applied to it, when you insert a keyframe at the opposite end of the tween, the movie clip automatically has the same filters, with the same stacking order, on the last frame of the tween as it did at the beginning of the tween.
- If you place movie clips on two different frames with different filters applied to each, and you apply a motion tween between the frames, Flash first processes the movie clip with the most filters. Flash then compares the filters applied to the first movie clip against the filters that the second movie clip uses. If no matching filters are found in the second movie clip, Flash generates a dummy filter with no parameters and the color of the existing filters.
- If a motion tween exists between two keyframes and you add a filter to the object in one keyframe, Flash automatically adds a dummy filter to the movie clip when it reaches the keyframe at the other end of the tween.
- If a motion tween exists between two keyframes and you remove a filter from an object in one keyframe, Flash automatically removes the matching filter from the movie clip when it reaches the keyframe at the other end of the tween.
- If you set filter parameters inconsistently between the beginning and end of a motion tween, Flash applies the filter settings of the starting frame to the interpolated frames. Inconsistent settings occur when the following parameters are set differently between the beginning and end of the tween: knockout, inner shadow, inner glow, and type of gradient glow and gradient bevel.

For example, if you create a motion tween using the drop shadow filter, and apply a drop shadow with a knockout on the first frame of the tween, and an inner shadow on the last frame of the tween, Flash corrects the inconsistent use of the filter in the motion tween. In this case, Flash applies the filter settings used on the first frame of the tween—a drop shadow with a knockout.

About filters and Flash Player performance

The type, number, and quality of the filters you apply to objects can affect the performance of SWF files as you play them. The more filters you apply to an object, the greater the number of calculations Adobe® Flash® Player must process to correctly display the visual effects you've created. Adobe® recommends that you apply a limited number of filters to a given object.

Each filter includes controls that let you adjust the strength and quality of the applied filter. Using lower settings improves performance on slower computers. If you are creating content for playback on a wide range of computers, or are unsure of the computing power available to your audience, set the quality level to Low to maximize playback performance.

About Pixel Bender filters

Adobe Pixel Bender™ is a programming language developed by Adobe that allows users to create custom filters, effects, and blend modes for use in Flash and After Effects. Pixel Bender is hardware independent and designed to run efficiently on a variety of GPU and CPU architectures automatically.

Pixel Bender developers create filters by writing Pixel Bender code and saving the code in a text file with the file extension `pbj`. Once written, a Pixel Bender filter can be used by any Flash document. Use ActionScript® 3.0 to load the filter and use its controls.

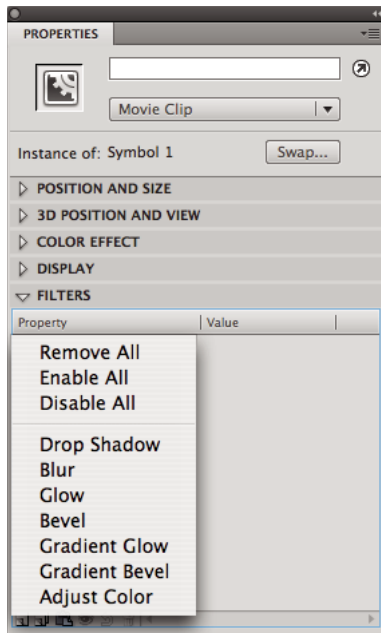
For more information about working with Pixel Bender in ActionScript, see *Programming ActionScript 3.0*.

Applying filters

Apply filters



Each time you add a new filter to an object, it is added to the list of applied filters for that object in the Property inspector. You can apply multiple filters to an object, as well as remove filters that were previously applied. You can apply filters only to text, button, and movie clip objects.

You can create a filter settings library that lets you easily apply the same filter or sets of filters to an object. Flash stores the filter presets you create in the Filters section of the Property inspector in the Filters > Presets menu.





The Add Filter menu in the Property inspector


Apply or remove a filter

- 1 Select a text, button, or movie clip object to apply a filter to or remove a filter from.
- 2 In the Filters section of the Property inspector, do one of the following:
 - To add a filter, click the Add Filter button , and select a filter. Experiment with the settings until you get the desired look.
 - To remove a filter, select the filter to remove in the list of applied filters, and click the Remove Filter button . You can delete or rename any presets.

Copy and paste a filter

- 1 Select the object to copy a filter from, and select the Filters panel.
- 2 Select the filter to copy, and click the Clipboard button  and choose Copy Selected from the pop-up menu. To copy all filters, choose Copy All.
- 3 Select the object to apply the filter to, and click the Clipboard button  and choose Paste from the pop-up menu.

Apply a preset filter to an object

- 1 Select the object to apply a filter preset to, and select the Filter tab.
- 2 Click the Add Filter button , and select Presets.
- 3 Select the filter preset to apply from the list of available presets at the bottom of the preset menu.


Note: When you apply a filter preset to an object, Flash replaces any filters currently applied to the selected objects with the filters used in the preset.

Enable or disable a filter applied to an object

- ❖ Click the enable or disable icon next to the filter name in the Filter list.

Note: To toggle the enable state of the other filters in the list, Alt-click (Windows) or Option-click (Macintosh) the enable icon in the Filter list. If you Alt-click the disable icon, the selected filter is enabled, and all others filters in the list are disabled.

Enable or disable all filters applied to an object

❖ Click the Add Filter button , and select Enable All or Disable All.



Note: To enable or disable all of the filters in the list, Control-click the enable or disable icon in the Filter list.

Create preset filter libraries



Save filter settings as preset libraries that you can easily apply to movie clip and text objects. Share your filter presets with other users by providing them with the filter configuration file. The filter configuration file is an XML file that is saved in the Flash Configuration folder in the following location:

- Windows XP: C:\Documents and Settings\username\Local Settings\Application Data\Adobe\Flash CS4\language\Configuration\Filters\filtername.xml
- Windows Vista: C:\Users\username\Local Settings\Application Data\Adobe\Flash CS4\language\Configuration\Filters\filtername.xml
- Macintosh: Macintosh HD/Users/username/Library/Application Support/Adobe/Flash CS4/language/Configuration/Filters/filtername.xml



Create a library of filters with preset settings

- 1 Apply the filter or filters to the object.
- 2 Click the Add Filter button , and add a new filter.
- 3 Select the filter and click the Presets menu , and choose Save As.
- 4 Enter a name for the filter settings in the Save Preset As dialog box, and click OK.

Rename a filter preset

- 1 Click the Add Filter button , and add a new filter.
- 2 Select the filter and click the Presets menu , and choose Rename.
- 3 Double-click the preset name to modify.
- 4 Enter a new preset name, and click Rename.

Delete a filter preset

- 1 Click the Add Filter button , and add a new filter.
- 2 Select the filter and click the Presets menu , and choose Delete.
- 3 Select the preset to remove, and click Delete.


Apply a drop shadow

The Drop Shadow filter simulates the look of an object casting a shadow onto a surface.

Text ...

Text with the Drop Shadow filter applied

For a sample of a drop shadow with a classic tween, see the Flash Samples page at www.adobe.com/go/learn_fl_samples. Download and decompress the Samples zip file and navigate to the Graphics\AnimatedDropShadow directory.

- 1 Select the object to apply a drop shadow to.
- 2 In the Filters section of the Property inspector, click the Add Filter button , and select Drop Shadow.
- 3 Edit the filter settings:
 - To set the width and height of the drop shadow, set the Blur X and Y values.
 - To set the darkness of the shadow, set the Strength value. The higher the numerical value, the darker the shadow.
 - Select the quality level for the drop shadow. High is approximate to that of a Gaussian blur. Low maximizes playback performance.
 - To set the angle of the shadow, enter a value.
 - To set the distance of the shadow from the object, set the Distance value.
 - Select Knockout to knock out (or visually hide) the source object and display only the drop shadow on the knockout image.
 - To apply the shadow within the boundaries of the object, select Inner shadow.
 - To hide the object and display only its shadow, select Hide Object. Hide Object lets you more easily create a realistic shadow.
 - To open the Color Picker and set the shadow color, click the Color control.

Create a skewed drop shadow



Skewing the Drop Shadow filter to create a more realistic looking shadow


- 1 Select the object with the shadow you want to skew.
- 2 Duplicate (select Edit > Duplicate) the source object.
- 3 Select the duplicated object, and skew it using the Free Transform tool (Modify > Transform > Rotate And Skew).
- 4 Apply the Drop Shadow filter to the duplicated movie clip or text object. (It will already be applied if the object you duplicated already had a drop shadow.)
- 5 In the Filters panel, select Hide Object to hide the duplicated object while leaving its shadow visible.
- 6 Select Modify > Arrange > Send Backward to place the duplicated object and its shadow behind the original object that you duplicated.
- 7 Adjust both the Drop Shadow filter settings and the angle of the skewed drop shadow until you achieve the desired look.

Apply a blur

The Blur filter softens the edges and details of objects. Applying a blur to an object can make it appear as if it is behind other objects, or make an object appear to be in motion.

Text ...

Text with the Blur filter applied


- 1 Select an object to apply a blur to, and select Filters.
- 2 Click the Add Filter button , and select Blur.
- 3 Edit the filter settings on the Filter tab:
 - To set the width and height of the blur, set the Blur X and Y values.
 - Select the quality level for the blur. High is approximate to that of a Gaussian blur. Low maximizes playback performance.

Apply a glow

The Glow filter lets you apply a color around the edges of an object.

Text ...

Text with the Glow filter applied

- 1 Select an object to apply a glow to, and select Filters.
- 2 Click the Add Filter button , and select Glow.
- 3 Edit the filter settings in the Filter tab:
 - To set the width and height of the glow, set the Blur X and Y values.
 - To open the Color Picker and set the glow color, click the Color control.
 - To set the sharpness of the glow, set the Strength value.
 - To knock out (or visually hide) the source object and display only the glow on the knockout image, select Knockout.

Text ...

Using the Glow filter with the Knockout option


- To apply the glow within the boundaries of the object, select Inner Glow.
- Select the quality level for the glow. High is approximate to that of a Gaussian blur. Low maximizes playback performance.

Apply a bevel

Applying a bevel applies a highlight to the object that makes it appear to be curved up above the background surface.

Text ...

Text with a bevel applied


- 1 Select an object to apply a bevel to, and select Filters.
- 2 Click the Add Filter button , and select Bevel.
- 3 Edit the filter settings in the Filter tab:
 - To set the type of bevel, select a bevel from the Type menu.
 - To set the width and height of the bevel, set the Blur X and Y values.
 - Select a shadow and highlight color for the bevel from the pop-up color palette.
 - To set the opacity of the bevel without affecting its width, set the Strength value.
 - To change the angle of the shadow that the beveled edge casts, set the Angle value.
 - To define the width of the bevel, enter a value for Distance.
 - To knock out (or visually hide) the source object and display only the bevel on the knockout image, select Knockout.

Apply a gradient glow

Applying a gradient glow produces a glow look with a gradient color across the surface of the glow. The gradient glow requires one color at the beginning of the gradient with an Alpha value of 0. You cannot move the position of this color, but you can change the color.

Text ...

Text with a gradient glow applied

- 1 Select an object to apply a gradient glow to.
- 2 In the Filters section of the Property inspector, click the Add Filter button , and select Gradient Glow.
- 3 Edit the filter settings on the Filter tab:
 - Select the type of glow to apply to the object from the Type pop-up menu.
 - To set the width and height of the glow, set the Blur X and Y values.
 - To set the opacity of the glow without affecting its width, set the Strength value.
 - To change the angle of the shadow that the glow casts, set the Angle value.
 - To set the distance of the shadow from the object, set the Distance value.
 - To knock out (or visually hide) the source object and display only the gradient glow on the knockout image, select Knockout.
 - Specify a gradient color for the glow. A gradient contains two or more colors that fade or blend into one another. The color you select for the beginning of the gradient is referred to as the *alpha* color.


To change a color in the gradient, select one of the color pointers below the gradient definition bar and click the color space that appears directly below the gradient bar to display the Color Picker. Sliding these pointers adjusts the level and position of that color in the gradient.

To add a pointer to the gradient, click on or below the gradient definition bar. To create a gradient with up to 15 color transitions, add up to 15 color pointers. To reposition a pointer on the gradient, drag the pointer along the gradient definition bar. To remove a pointer, drag it down and off the gradient definition bar.

- Select the quality level for the gradient glow. High is approximate to that of a Gaussian blur. Low maximizes playback performance.

Apply a gradient bevel

Applying a gradient bevel produces a raised look that makes an object appear to be raised above the background, with a gradient color across the surface of the bevel. The gradient bevel requires one color in the middle of the gradient with an alpha value of 0.


- 1 Select an object to apply a gradient bevel to.
- 2 In the Filters section of the Property inspector, click the Add Filter button , and select Gradient Bevel.
- 3 Edit the filter settings on the Filter tab:
 - Select the type of bevel to apply to the object from the Type pop-up menu.
 - To set the width and height of the bevel, set the Blur X and Y values.
 - To affect the smoothness of the bevel without affecting its width, enter a value for Strength.
 - To set the angle of the light source, enter a value for Angle.
 - To knock out (or visually hide) the source object and display only the gradient bevel on the knockout image, select Knockout.
 - Specify a gradient color for the bevel. A gradient contains two or more colors that fade or blend into one another. The middle pointer controls the alpha color of the gradient. You can change the color of the alpha pointer, but you cannot reposition this color in the gradient.

To change a color in the gradient, select one of the color pointers below the gradient definition bar, and click the color space that appears directly below the gradient bar to display the Color Picker. To adjust the level and position of that color in the gradient, slide these pointers.

To add a pointer to the gradient, click on or below the gradient definition bar. To create a gradient with up to 15 color transitions, add up to 15 color pointers. To reposition a pointer on the gradient, drag the pointer along the gradient definition bar. To remove a pointer, drag it down and off the gradient definition bar.

Apply the Adjust Color filter

The Adjust Color filter allows you to finely control the color attributes of the selected object, including contrast, brightness, saturation, and hue.

- 1 Select an object to adjust the color for.
- 2 In the Filters section of the Property inspector, click the Add Filter button , and select Adjust Color.
- 3 Enter values for the color attributes. The attributes and their corresponding values are as follows:

Contrast Adjusts the highlights, shadows, and midtones of an image.

Brightness Adjusts the brightness of an image.

Saturation Adjusts the intensity of a color.

Hue Adjusts the shade of a color.

- 4 To reset all of the color adjustments to 0 and return the object to its original state, click Reset Filter.

See also

“[Change the color and transparency of an instance](#)” on page 153

Applying blend modes

About blend modes

Blend modes let you create composite images. *Compositing* is the process of varying the transparency or color interaction of two or more overlapping objects. Blending lets you create unique effects by blending the colors in overlapping movie clips.

A blending mode contains the following elements:

Blend color The color applied to the blend mode.

Opacity The degree of transparency applied to the blend mode.

Base color The color of pixels underneath the blend color.

Result color The result of the blend’s effect on the base color.

Blend modes depend on both the color of the object you’re applying the blend to and the underlying color. Adobe® recommends that you experiment with the different blend modes to achieve the desired effect.

Normal Applies color normally, with no interaction with the base colors.

Layer Lets you stack movie clips on top of each other without affecting their color.

Darken Replaces only the areas that are lighter than the blend color. Areas darker than the blend color don’t change.

Multiply Multiplies the base color by the blend color, resulting in darker colors.

Lighten Replaces only pixels that are darker than the blend color. Areas lighter than the blend color don’t change.

Screen Multiplies the inverse of the blend color by the base color, resulting in a bleaching effect.

Overlay Multiplies or screens the colors, depending on the base colors.

Hard Light Multiplies or screens the colors, depending on the blend mode color. The effect is similar to shining a spotlight on the object.

Difference Subtracts either the blend color from the base color or the base color from the blend color, depending on which has the greater brightness value. The effect is similar to a color negative.

Add Commonly used to create an animated lightening dissolve effect between two images.

Subtract Commonly used to create an animated darkening dissolve effect between two images.

Invert Inverts the base color.

Alpha Applies an alpha mask.

Erase Removes all base color pixels, including those in the background image.

Note: Erase and Alpha blend modes require that a Layer blend mode be applied to the parent movie clip. You cannot change the background clip to Erase and apply it because the object would appear invisible.

Blend mode examples

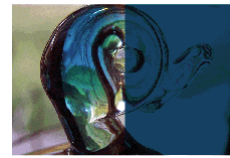
The following examples illustrate how different blend modes affect the appearance of an image. The resulting effect of a blend mode might be considerably different, depending on the color of the underlying image and the type of blend mode you apply.



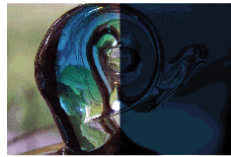
Original image



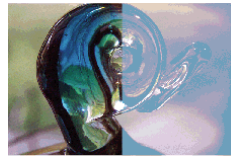
Layer



Darken



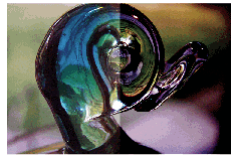
Multiply



Lighten



Screen



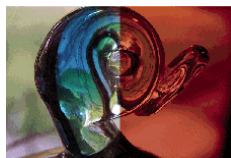
Overlay



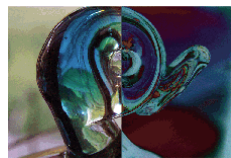
Hard Light



Add



Subtract



Difference



Invert

Apply a blend mode

To apply blends to selected movie clips, use the Property inspector.

Note: You cannot apply different blend modes to different graphic symbols because multiple graphic symbols are merged as a single shape when you publish the SWF file.

- 1 Select the movie clip instance (on the Stage) to apply a blend mode to.
- 2 To adjust the color and transparency of the movie clip instance, use the Color pop-up menu in the Properties panel.
- 3 Select a blend mode for movie clips from the Blend pop-up menu in the Properties panel. The blend mode is applied to the selected movie clip instance.
- 4 Verify that the blend mode you selected is appropriate to the effect you're trying to achieve.

Experiment with both the color and transparency settings of the movie clip and the different blend modes to achieve the desired effect.

See also

[“Change the color and transparency of an instance”](#) on page 153

Chapter 9: Text

This chapter describes the various ways that text can be incorporated into your Adobe® Flash® CS4 Professional documents.

About text and fonts

About text

You can include text in your Adobe® Flash® CS4 Professional applications in a variety of ways. You can create text fields containing *static* text, which you create when you author the document. You can also create *dynamic* text fields, which display updating text, such as stock quotes or news headlines, and *input* text fields, which allow users to enter text for forms or surveys.

Flash provides many ways to work with text. For example, you can orient text horizontally or vertically; set attributes such as font, size, style, color, and line spacing; check spelling; transform text by rotating, skewing, or flipping; link text; make text selectable; animate text; control font substitution; and use a font as part of a shared library. Flash documents can use Type 1 PostScript® fonts, TrueType®, and bitmap fonts (Macintosh only).

You can preserve rich text formatting in text fields, using HTML tags and attributes. When you use HTML text for the content of a dynamic or input text field, the text can flow around an image, such as a SWF or JPEG file or a movie clip. See Using HTML-formatted text in [Learning ActionScript 2.0 in Adobe Flash](#).

Like movie clip instances, text field instances are ActionScript® objects that have properties and methods. By giving a text field an instance name, you can manipulate it with ActionScript. However, you cannot write ActionScript code inside a text instance, because text instances don't have Timelines.

You can use ActionScript to format input and dynamic text, and to create scrolling text fields. ActionScript has events for dynamic and input text fields that you can capture and use to trigger scripts. For information on using ActionScript to control text, see Working with Text and Strings in [Learning ActionScript 2.0 in Adobe Flash](#).

See also

[“About text fields”](#) on page 246

Flash Text Anti-Aliasing

Anti-aliasing lets you smooth the edges of onscreen text. The anti-aliasing options are particularly effective for rendering smaller font sizes. When anti-aliasing is enabled, all text in the current selection is affected. Anti-aliasing operates with text of all point sizes in the same way.

Anti-aliasing is supported for static, dynamic, and input text if the user has Flash® Player 7 or later. It is supported only for static text if the user has an earlier version of Flash Player.

When using small text in a Flash document, keep in mind the following guidelines:

- Sans serif text, such as Helvetica or Arial, appears clearer at small sizes than serif text.
- Some type styles, such as bold and italic, can make text less legible at small sizes.
- In some cases, text appears somewhat smaller than text of the same point size in other applications.

The Flash text rendering engine provides clear, high-quality text rendering in Flash (FLA) documents and published SWF files. The Anti-alias for Readability setting makes text more legible, particularly at small font sizes. Custom anti-aliasing lets you specify the thickness and sharpness of fonts used in individual text fields.

High quality anti-aliasing is automatically enabled whenever you publish to Flash Player 8 or later and Anti-Alias For Readability or Custom Anti-Alias is selected. Anti-Alias For Readability may cause a slight delay when you load Flash SWF files, especially if you are using four or five different character sets in the first frame of a Flash document. High-quality anti-aliasing may also increase Flash Player's memory usage. Using four or five fonts, for example, can increase memory usage by approximately 4 MB.

When the publish setting of your file is Adobe® Flash® Player 8 or later, and Anti-Alias For Readability or Custom Anti-Alias is your chosen anti-aliasing option, high-quality anti-aliasing applies to the following:

- Untransformed text that is scaled or rotated

Note: Although the text can be scaled or rotated, it must remain flat (that is, untransformed). For example, if you skew the fonts or otherwise manipulate the font shapes, Anti-Alias for Readability is automatically disabled.

- All font families (including bold, italic, and so on)
- Display sizes of up to 255 points
- Exporting to most non-Flash file formats (GIF or JPEG)

High-quality anti-aliasing is disabled under the following conditions:

- Flash Player 7 or earlier is the selected version of Flash Player.
- An anti-aliasing option other than Anti-Alias for Readability or Custom Anti-Alias is selected.
- Text is skewed or flipped.
- The FLA file is exported to a PNG file.

See also

[“Work with text anti-aliasing”](#) on page 256

[“Font outlines and device fonts”](#) on page 245

Unicode text encoding in SWF applications

Flash Player 7 and later support Unicode text encoding for SWF files in Flash Player format. This support greatly enhances your ability to use multilanguage text in your SWF files, such as two languages within a single text field. Any user with Flash Player 7 or later can view multilanguage text in a Flash Player 7 or later application, regardless of the language used by the operating system running the player.

See also

[“Creating multilanguage text”](#) on page 263

Font outlines and device fonts

When you publish or export a Flash document, fonts in text fields are represented by embedded font outlines or by font names. Alternatively, you can use device fonts to specify a general type of font, such as sans serif.

Font outlines and names

For static text, Flash creates outlines of the font and embeds them in the SWF file. Flash Player then uses the outlines to display the text.

For dynamic or input text, Flash stores the font names; Flash Player then locates identical or similar fonts on the user's system when the Flash application is displayed. To ensure that users have the correct fonts for dynamic or input text, you can embed font outlines, but this can increase file size.

Not all fonts displayed in Flash can be exported as outlines with a Flash application. To verify that a font can be exported, use the View > Preview Mode > Antialias Text command to preview the text; jagged type indicates that Flash does not recognize that font's outline and will not export the text.

Device fonts

Use generic device fonts as an alternative to embedding font outline information. Flash includes three generic device fonts, `_sans` (similar to Helvetica or Arial), `_serif` (similar to Times Roman), and `_typewriter` (similar to Courier). When you specify one of these fonts and then export the document, Flash Player uses the font on the user's computer that most closely resembles the generic device font.

Because device fonts are not embedded, they result in a smaller SWF file. They can also result in more legible text at small point sizes (below 10 points). However, if a user's computer does not have an installed font corresponding to the device font, text may look different than expected.

See also

[“Create and edit text fields”](#) on page 248

[“Use device fonts”](#) on page 257

Masking device fonts

You can use a movie clip to mask device font text in another movie clip. (You cannot mask device fonts by using a mask layer on the Stage.) For this movie clip mask to function, the user must have Flash Player 6 (6.0.40.0) from Adobe or later.

When you use a movie clip to mask device font text, Flash uses the rectangular bounding box of the mask as the masking shape. That is, if you create a nonrectangular movie clip mask for device font text in the Flash authoring environment, the mask that appears in the SWF file takes the shape of the rectangular bounding box of the mask, not the shape of the mask itself.

For more information on using a movie clip as a mask, see Using movie clips as masks in [Learning ActionScript 2.0 in Adobe Flash](#).

For a sample of device font masking, see the Flash Samples web page at www.adobe.com/go/learn_fl_samples. Download and decompress the Samples zip file and navigate to the Masking\DeviceFontMasking folder to access the sample.

Creating text

About text fields

You can create three types of text fields: static, dynamic, and input. All text fields support Unicode.

- Static text fields display text that doesn't change characters dynamically.

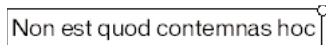
- Dynamic text fields display dynamically updating text, such as stock quotes or weather reports.
- Input text fields allow users to enter text in forms or surveys.

You can create horizontal text (with a left-to-right flow) or static vertical text (with either a right-to-left or left-to-right flow).

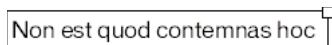
When creating static text, you can place text on a single line that expands as you type, or in a fixed-width field (for horizontal text) or fixed-height field (for vertical text) that expands and wraps words automatically. When creating dynamic or input text, you can place text on a single line, or create a text field with a fixed width and height.

Flash displays a handle on the corner of a text field to identify the type of text field:

- For static horizontal text that extends, a round handle appears at the upper-right corner of the text field.



- For static horizontal text that has a fixed width, a square handle appears at the upper-right corner of the text field.



- For static vertical text that has right-to-left flow and extends, a round handle appears at the lower-left corner of the text field.



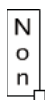
- For static vertical text that has right-to-left flow and a fixed height, a square handle appears at the lower-left corner of the text field.



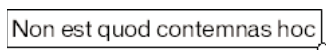
- For static vertical text that has left-to-right flow and extends, a round handle appears at the lower-right corner of the text field.



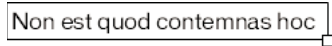
- For static vertical text that has left-to-right flow and a fixed height, a square handle appears at the lower-right corner of the text field.



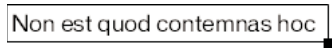
- For dynamic or input text fields that extend, a round handle appears at the lower-right corner of the text field.



- For dynamic or input text that has a defined height and width, a square handle appears at the lower-right corner of the text field.



- For dynamic scrollable text fields, the round or square handle becomes solid black instead of hollow.



Shift-double-click the handle of dynamic and input text fields to create text fields that don't expand when you enter text on the Stage. This allows you to create a text field of a fixed size and fill it with more text than it can display to create scrolling text.

After you use the Text tool to create a text field, use the Property inspector to specify the type of text field, and to set values that control how the text field and its contents appear in the SWF file.

Create and edit text fields

Text is horizontal by default. However, static text can also be aligned vertically.

You can use most common word-processing techniques to edit text in Flash. Use the Cut, Copy, and Paste commands to move text in a Flash file as well as between Flash and other applications.

See also

[“Setting text attributes”](#) on page 254


Add text to the Stage

- 1 Select the Text tool **T**.
- 2 In the Property inspector (Window > Properties), select a text type from the pop-up menu to specify the type of text field:

Dynamic Text Creates a field that displays dynamically updating text.

Input Text Creates a field in which users can enter text.

Static Text Creates a field that cannot update dynamically.

- 3 For static text only: In the Property inspector, click Change Orientation Of Text  and select a direction for text orientation and flow. (Horizontal is the default setting.)
- 4 On the Stage, do one of the following:
 - To create a text field that displays text in a single line, click where you want the text to start.
 - To create a text field with a fixed width (for horizontal text) or fixed height (for vertical text), position the pointer where you want the text to start and drag to the desired width or height.

Note: If you create a text field that extends past the edge of the Stage as you type, the text isn't lost. To make the handle accessible again, add line breaks, move the text field, or select View > Pasteboard.

- 5 Select text attributes in the Property inspector.

Change the size of a text field


- ❖ Drag the text field's resize handle.

When text is selected, a blue bounding box lets you resize the text field by dragging one of its handles. Static text fields have four handles that let you resize the text field horizontally. Dynamic text fields have eight handles that let you resize the text field vertically, horizontally, or diagonally.


Switch a text field between fixed-width (or fixed-height) and extending

- ❖ Double-click a resize handle.




Select characters in a text field

- 1 Select the Text tool .
- 2 Do one of the following:
 - Drag to select characters.
 - Double-click to select a word.
 - Click to specify the beginning of the selection, and Shift-click to specify the end of the selection.
 - Press Control+A (Windows) or Command+A (Macintosh) to select all the text in the field.

Select text fields

- ❖ Using the Selection tool , click a text field. Shift-click to select multiple text fields.

Set dynamic and input text options

- 1 Click in an existing dynamic text field.
- 2 In the Property inspector, make sure Dynamic or Input is displayed in the pop-up menu.
- 3 Enter an instance name for the text field.
- 4 Specify the height, width, and location of text.
- 5 Select the font and style.
- 6 In the Paragraph section of the Property inspector, specify one of the following options from the Behavior menu:
 - Single line** Displays the text as one line.
 - Multiline** Displays the text in multiple lines.
 - Multiline No Wrap** Displays text in multiple lines that break only if the last character is a breaking character, such as Enter (Windows) or Return (Macintosh).
- 7 To enable users to select dynamic text, click Selectable . Deselect this option to prevent users from selecting dynamic text.
- 8 To preserve rich text formatting (such as fonts and hyperlinks) with the appropriate HTML tags, click Render Text As HTML .
- 9 To display a black border and white background for the text field, click Show Border Around Text .
- 10 (Optional) In the Var box, enter the variable name for the text field. (Use this option only when authoring for Macromedia Flash Player 5 from Adobe or earlier.)

Beginning with Macromedia Flash MX (version 6), you assign the text field an instance name using the Property inspector. Although you can use the variable name method with dynamic text fields for backwards compatibility to Macromedia Flash 5 and earlier versions, Adobe doesn't recommend this, because you can't control other text field properties, or apply style sheet settings.

11 Click Character Embedding for embedded font outline options:

Don't embed Specifies that no fonts be embedded.

Auto fill Click Auto Fill to embed all of the characters from the selected text field.

Set preferences for vertical text

1 Select Edit > Preferences (Windows) or Flash > Preferences (Macintosh) and click the Text category in the Preferences dialog box.

2 Under Vertical Text, set any of these options:

Default Text Orientation Automatically gives new text fields vertical orientation.

Right to Left Text Flow Makes lines of vertical text fill the page from right to left.

No Kerning Prevents kerning from being applied to vertical text. (Kerning remains enabled for horizontal text.)

Check spelling

You can check spelling in text throughout your Flash document. You can also customize the spell checker.

Use the spell checker

1 Select Text > Check Spelling to view the Check Spelling dialog box.

The box in the upper-left corner identifies words not found in the selected dictionaries, and also identifies the type of element that contains the text (such as a text field or frame label).

2 Do one of the following:

- Click Add To Personal to add the word to your personal dictionary.
- Click Ignore to leave the word unchanged. Click Ignore All to leave all occurrences of the word in the document unchanged.
- Enter a word in the Change To box or select a word from the Suggestions scroll list. Then click Change to change the word or click Change All to change all occurrences of the word in the document.
- Click Delete to delete the word from the document.

3 To end the spelling check, do one of the following:

- Click Close to end spelling before Flash reaches the end of the document.
- Continue checking spelling until you see a notification that Flash has reached the end of the document, then click No to end spelling checking. (Click Yes to continue the spelling check at the beginning of the document.)

Customize the spelling checker

1 Do one of the following:

- Select Text > Spelling Setup. (Use this option if you have not used the Check Spelling feature before.)
- In the Check Spelling dialog box (Text > Check Spelling), click Setup.

2 Set any of the following options:

Document Options Use these options to specify which elements are to be checked.

Dictionaries Lists the built-in dictionaries. You must select at least one dictionary to enable spelling checking.

Personal Dictionary Enter a path or click the folder icon and browse to a document to use as a personal dictionary. (You can modify this dictionary.)

Edit Personal Dictionary Adds words and phrases to your personal dictionary. In the Personal Dictionary dialog box, enter each new item on a separate line in the text field.

Checking Options Use these options to control how Flash handles specific types of words and characters when checking spelling.

Embed and share fonts

You can embed a font in your SWF file so that the font does not need to be present on the devices the SWF file eventually plays back on. To embed a font, create a font library item.

Creating a font library item also allows you to use the font as a shared library item for sharing among multiple SWF files. You must also assign the font item a linkage identifier string and a URL where the document that contains the font symbol will be posted. In this way, you can link to the font and use it in a Flash application. When you use font symbols for dynamic or input text, embed the font outline information.

After a font symbol in a Flash document has an assigned identifier string and URL, use the font symbol in another Flash document by copying the font symbol into the destination FLA file.

See also

[“Sharing library assets”](#) on page 162

[“Copy library assets between documents”](#) on page 160

Create a font library item

- 1 Open the library to add a font symbol to.
- 2 Select New Font from the Library Panel menu.
- 3 Enter a name for the font item in the Name text field.
- 4 Select a font family from the Font menu or enter the name of a font in the Font text field.
- 5 (Optional) Select a font style from the Style menu.

If the selected font does not include a bold or italic style, you can select the Faux Bold or Faux Italic checkboxes. Faux Bold and Faux Italic styles are added to the Regular style by the operating system. The faux styles may not look as good as fonts that include a true bold or italic style.

- 6 (Optional) To embed the font information as bitmap data rather than vector outline data, select the Bitmap Text option, and enter a font size in the Size text field. (Bitmap fonts cannot use anti-aliasing. You must choose Bitmap as the anti-aliasing option in the Property inspector for text that uses this font.)

Note: The Size setting applies only when you use the Bitmap Text option.

Assign an identifier string to a font library item

- 1 Select the font item in the Library panel.
- 2 Select Properties from the Library Panel menu.
- 3 Under Linkage, select Export for Runtime Sharing.
- 4 In the Identifier text field, enter a string to identify the font item.
- 5 In the URL text field, enter the URL of the SWF file that contains the font item.


Scrolling text

There are several ways to create scrolling text in Flash:

- Make dynamic or input text fields scrollable by using menu commands or the text field handle. This does not add a scrollbar to the text field, but instead allows the user to scroll the text with the arrow keys (for text fields also set to Selectable) or the mouse wheel. The user must first click the text field to give it focus.
- Add a ScrollBar component to a text field to make it scroll. For more information, see “UIScrollBar Component” in the *ActionScript 2.0 Components Language Reference*.
- In ActionScript, use the TextField object’s `scroll` and `maxscroll` properties to control vertical scrolling and the `hscroll` and `maxhscroll` properties to control horizontal scrolling in a text field. See Example: Creating scrolling text in *Learning ActionScript 2.0 in Adobe Flash*.

Make dynamic text scrollable

❖ Do one of the following:

- Shift-double-click the lower-right handle on the dynamic text field. The handle will turn from an unfilled square (non-scrollable) to a filled square (scrollable).
- Using the Selection tool , select the dynamic text field and then select Text > Scrollable.
- Select the dynamic text field with the Selection tool. Right-click (Windows) or Control-click (Macintosh) the dynamic text field and select Scrollable from the context menu.

Transforming text

You can create text effects by transforming text fields. For example, you can rotate, skew, flip, and scale text fields. (When you scale a text field as an object, the Property inspector does not reflect increases or decreases in point size.) The text in a transformed text field can still be edited, although severe transformations may make it difficult to read.

You can also animate text by using Timeline effects. For example, you can make text bounce, fade in or out, or explode.

See also

“Transforming objects” on page 113


“Reshaping objects” on page 111

Break text apart

You can break apart text to place each character in a separate text field. Then you can quickly distribute the text fields to separate layers and animate each field. (You cannot break apart text in scrollable text fields.)

You can also convert the text to its component lines and fills to reshape, erase, and otherwise manipulate it. As with any other shape, you can individually group these converted characters, or change them to symbols and animate them. After you convert text to lines and fills, you can no longer edit the text.

Note: The Break Apart command applies only to outline fonts such as TrueType fonts. Bitmap fonts disappear from the screen when you break them apart. PostScript fonts can be broken apart only on Macintosh systems.

- 1 Using the Selection tool , click a text field.
- 2 Select Modify > Break Apart.

Each character in the selected text is placed in a separate text field. The text remains in the same position on the Stage.


- 3 Select Modify > Break Apart again to convert the characters to shapes on the Stage.

See also

“[Distributing objects to layers for tweened animation](#)” on page 186

“[Timelines and Animation](#)” on page 172

Link horizontal text to a URL

- 1 Select text or a text field:
 - Use the Text tool **T** to select text in a text field.
 - To link all the text in a text field, use the Selection tool  to select a text field.
- 2 In the Link text field in the Options section of the Property inspector (Window > Properties), enter the URL to which you want to link the text field.

Note: To create a link to an e-mail address, use the `mailto:` URL. For example, enter `mailto:adamsmith@example.com`.

Substituting missing fonts

If you work with a document that contains fonts that aren't installed on your system, Flash uses the fonts available on your system. You can select which fonts are substituted for the missing fonts, or you can let Flash substitute missing fonts with the Flash System Default Font.

When a font is substituted, the text is displayed on your system using the substitute font, but the missing font information is saved with the document. If the document is reopened on a system that includes the missing font, the text is displayed in that font.

When you work with a document that includes missing fonts, the missing fonts appear in the font list in the Property inspector. When you select substitute fonts, the substitute fonts names also appear, enclosed in parentheses to identify them as substitute fonts.

If you apply formatting (such as font size or kerning) to the substitute font, check the formatting when the text is displayed in the missing font, because it may not look as expected.

If you install a previously missing font on your system and restart Flash, the font is displayed in documents that use the font, and the font is removed from the Missing Fonts dialog box.

If you see an alert box indicating missing fonts in a document, you can select substitute fonts in the Font Mapping dialog box.

Specify font substitution

- 1 When the Missing Fonts alert appears, do one of the following:
 - Click Select Substitute Fonts to select substitute fonts from fonts installed on your system and proceed to step 2.
 - Click Use Default to use the Flash System Default Font to substitute all missing fonts and to dismiss the Missing Fonts alert. You have finished specifying font substitution.
- 2 In the Font Mapping dialog box, click a font in the Missing Fonts column to select it. Shift-click to select multiple missing fonts to map them all to the same substitute font.

The default substitute fonts are displayed in the Mapped To column, until you select substitute fonts.

- 3 Select a font from the Substitute Font pop-up menu.
- 4 Repeat steps 2 through 3 for all missing fonts.

View all the missing fonts in a document and reselect substitute fonts

- 1 With the document active in Flash, select Edit > Font Mapping (Windows) or Flash > Font Mapping (Macintosh).
- 2 Click a font in the Missing Fonts column to select it. Shift-click to select multiple missing fonts to map them all to the same substitute font.

The default substitute fonts are displayed in the Mapped To column, until you select substitute fonts.

- 3 Select a font from the Substitute Font pop-up menu.
- 4 Repeat steps 2 through 3 for all missing fonts.

View or delete font mappings

- 1 Close all documents in Flash.
- 2 Select Edit > Font Mapping.
- 3 To delete a font mapping, select the mapping and press Delete.

Turn off the Missing Fonts alert

- To turn the alert off for the current document, in the Missing Fonts alert box select Don't Show Again For This Document, Always Use Substitute Fonts. Select Edit > Font Mapping to view mapping information for the document again.

Setting text attributes

About text attributes

You can set the font and paragraph attributes of text. Font attributes include font family, point size, style, color, letter spacing, autokerning, and character position. Paragraph attributes include alignment, margins, indents, and line spacing.

For static text, font outlines are exported in a published SWF file. For horizontal static text, you can use device fonts instead of exporting font outlines.

For dynamic or input text, Flash stores the names of the fonts, and Flash Player locates identical or similar fonts on the user's system. You can also embed font outlines in dynamic or input text fields. Embedding font outlines can increase file size, but it ensures that users have the correct font information.


When creating new text, Flash uses the text attributes that are currently set in the Property inspector. When you select existing text, use the Property inspector to change font or paragraph attributes, and to direct Flash to use device fonts rather than embedding font outline information.

See also

[“Font outlines and device fonts”](#) on page 245

[“Create and edit text fields”](#) on page 248

Set a font, point size, style, and color

- 1 Using the Selection tool , select one or more text fields on the Stage.
- 2 In the Property inspector (Window > Properties), select a font from the Family pop-up menu, or enter a font name.

Note: The `_sans`, `_serif`, `_typewriter`, and device fonts can be used only with static horizontal text.

3 Enter a value for the font size.


Font size is set in points, regardless of the current ruler units.

4 To apply bold or italic style, select the style from the Style menu.

If the selected font does not include a bold or italic style, the style does not appear in the menu. You can select the Faux Bold or Faux Italic styles from the Text menu (Text > Style > Faux Bold or Faux Italic). Faux Bold and Faux Italic styles are added to the Regular style by the operating system. The faux styles may not look as good as fonts that include a true bold or italic style.

5 Select a font rendering method from the Anti-Aliasing pop-up menu (directly below the Color control) to optimize text.

6 To select a fill color for text, click the Color control and do one of the following:

- Select a color from the Color menu.
- Type a color's hexadecimal value in the box in the upper-left corner.
- Click Color Picker  and select a color from the system color picker. (When setting the text color, use only solid colors, not gradients. To apply a gradient to text, break the text apart and convert the text to its component lines and fills.)

See also


[“Font outlines and device fonts”](#) on page 245

Set letter spacing, kerning, and character position

Letter spacing inserts a uniform amount of space between characters. Use letter spacing to adjust the spacing of selected characters or entire blocks of text.

Kerning controls the spacing between pairs of characters. Many fonts have built-in kerning information. For example, *A* and *V* are often closer together than *A* and *D*. Flash provides horizontal tracking and kerning (for horizontal text) and vertical tracking and kerning (for vertical text).

For vertical text, you can disable kerning by default in Flash Preferences. If you do this and leave the kerning option selected in the Property inspector, kerning is applied to horizontal text only.

1 Using the Text tool , select one or more sentences, phrases, or text fields on the Stage.

2 In the Property inspector (Window > Properties), set the following options:

- To specify letter spacing (tracking and kerning), enter a value in the Letter Spacing field.
- To use a font's built-in kerning information, select Auto-Kern.
- To specify superscript or subscript character position, click the Toggle Superscript or Toggle Subscript button. The default position is Normal. Normal places text on the baseline, Superscript places text above the baseline (horizontal text) or to the right of the baseline (vertical text), and Subscript places text below the baseline (horizontal text) or to the left of the baseline (vertical text).

Set alignment, margins, indents, and line spacing

Alignment determines the position of each line of text in a paragraph relative to edges of the text field. Horizontal text is aligned relative to the left and right edges of the text field, and vertical text is aligned relative to the top and bottom edges of the text field. Text can be aligned to one edge of the text field, centered in the text field, or aligned to both edges of the text field (full justification).

Margins determine the amount of space between the border of a text field and its text. Indents determine the distance between the margin of a paragraph and the beginning of the first line.

Line spacing determines the distance between adjacent lines in a paragraph. For vertical text, line spacing adjusts the space between vertical columns.

Work with horizontal text

- 1 Using the Text tool **T**, select one or more text fields on the Stage.
- 2 In the Property inspector (Window > Properties), set the following options:
 - To set alignment, click Left, Center, Right, or Full Justification.
 - To set the left or right margin, enter values in the Margins text fields in the Paragraph section of the Property inspector.
 - To specify indents, enter a value in the Indentation text field in the Paragraph section of the Property inspector.
 - To specify line spacing, enter a value in the Line Spacing text field in the Paragraph section of the Property inspector.

Work with vertical text

- 1 Using the Text tool **T**, select one or more text fields on the Stage.
- 2 In the Property inspector (Window > Properties), set the following options:
 - To set alignment, click Top, Center, Bottom, or Full Justification.
 - To set the top or bottom margin, enter values in the Margins fields in the Paragraph section of the Property inspector.
 - To specify indents, enter a value in the Indentation text field in the Paragraph section of the Property inspector.
 - To specify line spacing, enter a value in the Line Spacing text field in the Paragraph section of the Property inspector.

Work with text anti-aliasing

Flash provides improved font rasterization that lets you specify the anti-aliasing properties for fonts. The improved anti-aliasing capabilities are available only for SWF files published for Flash Player 8 or later. If you are publishing files for earlier versions of Flash Player, you can only use the Anti-Alias For Animation feature.

Apply anti-aliasing for each text field rather than each character. Also, when you open existing FLA files in Flash 8 or later, the text is not automatically updated to the advanced anti-aliasing options; you must select individual text fields and manually change the anti-aliasing settings.

When you open a FLA file created for use with Flash Player 7 or earlier, the text Property inspector sets the anti-alias option to its equivalent anti-aliasing option from Flash MX 2004. However, to use the advanced anti-aliasing features, you must upgrade the FLA content for Flash Player 8 or later.

Choose an anti-aliasing option for selected text

- ❖ In the Property inspector, choose one of the following options from the Anti-Aliasing pop-up menu:

Use Device Fonts Specifies that the SWF file use the fonts installed on the local computer to display the fonts.

Typically, device fonts are legible at most font sizes. Although this option doesn't increase the size of the SWF file, it forces you to rely on the fonts installed on the user's computer for font display. When using device fonts, choose only commonly installed font families.

Bitmap Text (No Anti-Alias) Turns off anti-aliasing and provides no text smoothing. The text is displayed using sharp edges, and the resulting SWF file size is increased because the font outlines are embedded in the file. Bitmap text is sharp at the exported size, but scales poorly.

Anti-Alias For Animation Creates a smoother animation by ignoring alignment and kerning information. This option creates a larger SWF file because font outlines are embedded. For legibility, use 10-point or larger type when specifying this option.

Anti-Alias For Readability Uses the Flash text rendering engine to improve the legibility of fonts, particularly at small sizes. This option creates a larger SWF file because font outlines are embedded. To use this option, you must publish to Flash Player 8 or later. (Do not use this option if you intend to animate text; instead, use Anti-Alias For Animation.)



Custom Anti-Alias Lets you modify the font's properties. Use Sharpness to specify the smoothness of the transition between the text edges and the background. Use Thickness to specify how thick the font anti-aliasing transition appears. (Larger values cause the characters to look thicker.) Specifying Custom Anti-Alias creates a larger SWF file because font outlines are embedded. To use this option, you must publish to Flash Player 8 or later.

Upgrade content for Flash 8 or later anti-aliasing

- 1 Open a FLA file created for use with Flash Player 7 or earlier.
- 2 In the Publish Settings dialog box (File > Publish Settings), select Flash Player 8 or Flash Player 9 from the Version pop-up menu.
- 3 Select the text field to apply the Anti-Alias For Readability or Custom Anti-Alias option to.
- 4 In the Property inspector, select Anti-Alias For Readability or Custom Anti-Alias from the Font Rendering Method pop-up menu.


Make text selectable

Static horizontal text or dynamic text can be selectable by users viewing your Flash application. (Input text is selectable by default.) After selecting text, the user can copy, cut, and then paste the text into a separate document.

- 1 Using the Text tool , select the horizontal text that you want to make selectable.
- 2 In the Property inspector (Window > Properties), select Static Text or Dynamic Text.
- 3 Click Selectable .

Use device fonts

You can use generic device fonts in text.

- 1 Using the Selection tool , select one or more text fields.
- 2 In the Property inspector (Window > Properties), select Static Text from the pop-up menu.
- 3 In the Font pop-up menu, select a device font:

_sans A font similar to Helvetica or Arial.

_serif A font similar to Times Roman.

_typewriter A font similar to Courier.

See also

[“Font outlines and device fonts”](#) on page 245

[“Masking device fonts”](#) on page 246

About multilanguage text

You can configure a FLA file to display text in different languages depending on the language of the operating system that plays the Flash content.

Multilanguage text in Flash

You can include multilanguage text in your document in the following ways:

- The Strings panel lets localizers edit strings in a central location in Flash or in external XML files with their preferred software or translation memory. Flash supports multiline strings in both the Strings panel and the XML files.
- Select which character sets to embed in your applications, which limits the number of character glyphs in your published SWF file and reduces its size.
- Use a Western-style keyboard to create text on the Stage in Chinese, Japanese, and Korean.
- If you have Unicode fonts installed on your system, enter text directly into a text field. Because the fonts are not embedded, your users must also have Unicode fonts.

Other, less common methods of including multilingual text in your Flash documents include the following:

- Include an external text file in a dynamic or input text field by using the `#include` action.
- Load external text or XML files into a Flash application at runtime by using the `loadVariables` or `getURL` actions, or the `LoadVars` or `XML` objects.
- Enter Unicode escape characters in the string value for a dynamic or input text field variable.
- Create an embedded font as a symbol in your Library.

For Unicode-encoded text to appear correctly, users must have access to fonts containing the glyphs (characters) used in that text.

For a sample of multilingual content, see the Flash Samples page at www.adobe.com/go/learn_fl_samples. Download and decompress the Samples zip file and navigate to the Text\MultilingualContent folder to access the sample.

See also

[“Workflow for authoring multilanguage text with the Strings panel”](#) on page 263

[“Creating multilanguage documents by using text variables”](#) on page 272

[“Multilanguage text and ActionScript”](#) on page 270

[“Create multilanguage documents using the #include action”](#) on page 271

[“Use ActionScript to load external files”](#) on page 270

[“Non-Unicode external files”](#) on page 261

[“Embed and share fonts”](#) on page 251

About fonts for Unicode-encoded text

When you use external files that are Unicode encoded, your users must have access to fonts containing all the glyphs used in your text files. By default, Flash stores the names of fonts used in dynamic or input text files. During SWF file playback, Flash Player 7 (and earlier versions) looks for those fonts on the operating system running the player.

If the text in a SWF file contains glyphs that the specified font does not support, both Flash Player 7 and Flash Player 8 attempt to locate a font on the user's system that supports those glyphs. The player cannot always locate an appropriate font. This behavior depends on the fonts available on the user's system, as well as on the operating system running Flash Player.

Embed fonts for dynamic or input text fields

For dynamic or input text fields, embed fonts. However, some fonts, particularly those used for Asian languages, can add significantly to the SWF file size when embedded. With Flash, you can select ranges of fonts to embed.

You can also embed fonts by creating a font symbol in the library.

See also

[“Embed and share fonts”](#) on page 251

Select and embed a range of fonts

- 1 On the Stage, select a dynamic or input text field, show the Property inspector (Window > Properties), and click Character Embedding.
- 2 Select the range of font sets to embed by doing any of the following:
 - Click a character set in the Character Set text field.
 - Select multiple ranges by Shift-clicking the first and last fonts of a contiguous range of fonts, or by Control-clicking (Windows) or Command-clicking (Macintosh) to select noncontiguous fonts, click Auto Fill, and then click OK.

Note: Select only the font sets to embed, so you do not exceed the internal maximum number of glyphs for the authoring tool (approximately 65,500). Flash does not perform error-checking to confirm that the selected character set contains glyphs. During the publish or export procedure, only glyphs that are present in the font are embedded in the SWF file.

Embed font sets from text on the Stage

- 1 On the Stage, select a dynamic or input text field, show the Property inspector (Window > Properties), and click Character Embedding.
- 2 Select the character sets to embed and click AutoFill.

Remove embedded font sets

- 1 On the Stage, select a dynamic or input text field, show the Property inspector (Window > Properties), and click Character Embedding.
- 2 Click Don't Embed to remove all of the embedded fonts.
- 3 Click OK to close the dialog box, or select other fonts sets to embed in the SWF file.

XML font embedding table

When you select ranges of fonts to embed in a FLA file, Flash uses the UnicodeTable.xml file to determine which characters to embed. The UnicodeTable.xml file contains ranges of characters required for various languages and resides in the user configuration folder of your computer. The is located in the following directories:

- Windows: <boot drive>\Documents and Settings\<user>\Local Settings\Application Data\Adobe\Flash<version>\<language>\Configuration\FontEmbedding\
- Macintosh: <user>/Library/Application Support/Adobe/Flash<version>/<language>/Configuration/FonEmbedding/

The font set groupings are based on the Unicode Blocks as defined by the Unicode Consortium. To provide a simpler workflow, when you select a particular language, all related glyph ranges are embedded even if they are scattered into disjointed groupings.

For example, if you select Korean, the following Unicode character ranges are embedded:

3131-318E Hangul symbols

3200-321C Hangul specials

3260-327B Hangul specials

327F-327F Korean symbol

AC00-D7A3 Hangul symbols

If you select Korean + CJK, a larger font set is embedded:

3131-318E Hangul symbols

3200-321C Hangul specials

3260-327B Hangul specials

327F-327F Korean symbol

4E00-9FA5 CJK symbols

AC00-D7A3 Hangul symbols

F900-FA2D CJK symbols

The following table gives more details about the font selections for embedded fonts:

Range	Description
Uppercase [A-Z]	Basic Latin uppercase glyphs
Lowercase [a-z]	Basic Latin lowercase glyphs
Numerals [0-9]	Basic Latin numeral glyphs
Punctuation [!@#%...]	Basic Latin punctuation
Basic Latin	Basic Latin glyphs within the Unicode range 0x0021 to 0x007E
Japanese Kana	Hiragana and Katakana glyphs (including half-width forms)
Japanese Kanji – Level 1	Japanese Kanji characters
Japanese (All)	Japanese Kana and Kanji (including punctuation and special characters)
Basic Hangul	Most commonly used Korean characters, Roman characters, punctuations, and special characters/symbols
Hangul (All)	11,720 Korean characters (sorted by Hangul syllables), Roman characters, punctuations, and special characters/symbols
Traditional Chinese – Level 1	5000 most commonly used Traditional Chinese characters used in Taiwan
Traditional Chinese (All)	All Traditional Chinese characters used in Taiwan and Hong Kong, and punctuations
Simplified Chinese – Level 1	6000 most commonly used Simplified Chinese characters used in mainland of China and punctuations
Chinese (All)	All Traditional and Simplified Chinese characters and punctuations

Range	Description
Thai	All Thai glyphs
Devanagari	All Devanagari glyphs
Latin I	Latin-1 Supplement range 0x00A1 to 0x00FF (including punctuation, superscripts and subscripts, currency symbols, and letter-like symbols)
Latin Extended A	Latin Extended-A range 0x0100 to 0x01FF (including punctuation, superscripts and subscripts, currency symbols, and letter-like symbols)
Latin Extended B	Latin Extended-B range 0x0180 to 0x024F (including punctuation, superscripts and subscripts, currency symbols, and letter-like symbols)
Latin Extended Add'l	Latin Extended Additional range 0x1E00 to 0x1EFF (including punctuation, superscripts and subscripts, currency symbols, and letter-like symbols)
Greek	Greek and Coptic, plus Greek Extended (including punctuation, superscripts and subscripts, currency symbols, and letter-like symbols)
Cyrillic	Cyrillic (including punctuation, superscripts and subscripts, currency symbols, and letter-like symbols)
Armenian	Armenian plus ligatures
Arabic	Arabic plus Presentation Forms-A and Presentation Forms-B
Hebrew	Hebrew plus Presentation Forms (including punctuation, superscripts and subscripts, currency symbols, and letter-like symbols)

Non-Unicode external files

If you load external text or XML files that are not Unicode-encoded into a Flash Player 7 application, the text in the external files does not appear correctly when Flash Player attempts to show them as Unicode. To tell Flash Player to use the traditional code page of the operating system that is running the player, add the following code as the first line of code in the first frame of the Flash application that is loading the data:

```
system.useCodepage = true;
```

Set the `system.useCodepage` property only once in a document; do not use it multiple times in a document to make the player interpret some external files as Unicode and some as other encoding, because this can yield unexpected results.

If you set the `system.useCodepage` property to `true`, the traditional code page of the operating system running the player must include the glyphs used in your external text file for the text to appear. For example, if you load an external text file that contains Chinese characters, those characters do not appear on a system that uses the CP1252 code page, because that code page does not include Chinese characters. To ensure that users on all platforms can view external text files used in your Flash applications, encode all external text files as Unicode and leave the `system.useCodepage` property set to `false` by default. This causes Flash Player to interpret the text as Unicode. For more information, see `useCodepage` (System.useCodepage property) in the [ActionScript 2.0 Language Reference](#).

Text encoding

All text in a computer is encoded as a series of bytes. Many different forms of encoding (and therefore, different bytes) represent text. Different kinds of operating systems use different kinds of encoding for text. For example, Western Windows operating systems usually use CP1252 encoding; Western Macintosh operating systems usually use MacRoman encoding; Japanese Windows and Macintosh systems usually use Unicode encoding.

Unicode can encode most languages and characters used throughout the world. The other forms of text encoding that computers use are subsets of the Unicode format, tailored to specific regions of the world. Some of these forms are compatible in some areas and incompatible in other areas, so using the correct encoding is critical.

Unicode has several forms. Flash Player versions 6 and 7 and later support text or external files in the 8-bit Unicode format UTF-8, and in the 16-bit Unicode formats UTF-16 BE (Big Endian) and UTF-16 LE (Little Endian).

Unicode and Flash Player

Flash Player 6 and later versions support Unicode text encoding. Users with Flash Player 6 or later can view multilanguage text, regardless of the language that the operating system running the player uses, if they have the correct fonts installed.

Flash Player assumes that all external text files associated with a Flash Player application are Unicode encoded, unless you tell the player otherwise.

For Flash applications in Flash Player 5 or earlier that are authored in Flash MX or earlier, Flash Player 6 and earlier versions display the text by using the traditional code page of the operating system running the player.

For background information on Unicode, see Unicode.org.

Text encoding in Flash Player

By default, Flash Player 7 and later assumes that all text it encounters is Unicode encoded. If your document loads external text or XML files, the text in these files should be UTF-8 encoded. Create these files by using the Strings panel or using a text or HTML editor that can save the files in Unicode format.

Unicode encoding formats that Flash Player supports

When reading text data in Flash, Flash Player looks at the first two bytes in the file to detect a byte order mark (BOM), a standard formatting convention used to identify the Unicode encoding format. If no BOM is detected, the text encoding is interpreted as UTF-8 (an 8-bit encoding format). It is recommended that you use UTF-8 encoding in your applications.

If Flash Player detects either of the following BOMs, the text encoding format is interpreted as follows:

- If the first byte of the file is 0xFE and the second is 0xFF, the encoding is interpreted as UTF-16 BE (Big Endian). This is used for Macintosh operating systems.
- If the first byte of the file is 0xFF and the second is 0xFE, the encoding is interpreted as UTF-16 LE (Little Endian). This is used for Windows operating systems.

Most text editors that can save files in UTF-16BE or LE automatically add the BOMs to the files.

Note: If you set the `system.useCodepage` property to `true`, the text is interpreted using the traditional code page of the operating system that is running the player; it is not interpreted as Unicode.

Encoding in external XML files

You cannot change the encoding of an XML file by changing the encoding tag. Flash Player identifies the encoding of an external XML file using the same rules as for all external files. If no BOM is encountered at the beginning of the file, the file is assumed to be in UTF-8 encoding. If a BOM is encountered, the file is interpreted as UTF-16BE or LE.

See also

[“Non-Unicode external files”](#) on page 261

Creating multilanguage text

You can configure a FLA file to display text in different languages depending on the language of the operating system that plays the Flash content.

Workflow for authoring multilanguage text with the Strings panel

The Strings panel lets you create and update multilingual content. You can specify content for text fields that span multiple languages, and have Flash automatically determine the content that should appear in a certain language based on the language of the computer running Flash Player.

The following steps describe the general workflow:

1. Author a FLA file in one language.

Any text to enter in another language must be in a dynamic or input text field.

2. In the Strings Panel Settings dialog box, select the languages to include and designate one of them as the default language.

A column for the language is added to the Strings panel. When you save, test, or publish the application, a folder with an XML file is created for each language.

3. In the Strings panel, encode each text string with an ID.

4. Publish the application.

A folder is created for each language you select, and within each language folder is an XML file for that language.

5. Send the published FLA file and XML folders and files to your translators.

Author in your native language and let the translators make the translation. They can use translation software directly in the XML files or in the FLA file.

6. When you receive the translations from your translators, import the translated XML files back into the FLA file.

Select and remove languages for translation

As many as 100 languages can appear on the Stage and in the Strings panel for translation. Each language you select becomes a column in the Strings panel. To show the text on the Stage in any of the languages you selected, change the Stage language. The selected language appears when you publish or test the file.

When selecting languages, use any of the languages provided in the menu, as well as any other Unicode-supported language.

See also

[“Publishing overview”](#) on page 379

Select a language

- 1 Select Window > Other Panels > Strings, and click Settings.
- 2 Add a language by doing one of the following:
 - In the Languages box, highlight a language to select, and click Add.

- If the language does not appear in the Languages box, in the blank field below the Languages box, type a language code in the format *xx*. (The language code is from ISO 639-1.) Click Add.
- 3 Repeat step 3 until you have added all the necessary languages.
 - 4 Select a default language from the Default runtime language menu. This language appears on systems that do not have one of the active languages you selected.
 - 5 To load an XML file for the languages from a different URL at runtime, type the URL in the URL text field and click OK. A column for each selected language appears in the Strings panel. The columns appear in alphabetical order.
 - 6 Save the FLA file. When you save the FLA file, a folder for each language you selected is created in the same folder indicated in the SWF publish path. If no SWF publish path has been selected, it is created in the folder the FLA file resides in. Within each language file an XML file is created that is used to load translated text.

Remove a language

- 1 Select Window > Other Panels > Strings, and click Settings.
- 2 In the Active languages field, highlight a language and click Remove.
- 3 Repeat step 3 until you have removed all the unwanted languages.
- 4 When you finish removing languages, click OK.

The column for each removed language no longer appears in the Strings panel.

Note: When you remove a language from the Strings panel, the language XML file is not deleted from the local file system. This lets you add the language back into the Strings panel by using the previous XML file, and prevents accidental deletion. To completely remove the language, you must delete or replace the language XML file.

Add strings to the Strings panel

Assign text strings to the Strings panel in the following ways:

- Assign a string ID to a dynamic or input text field
- Add a string to the Strings panel without assigning it to a text field
- Assign an existing string ID to an existing dynamic or input text field

See also

[“Creating text”](#) on page 246

Assign a string ID to a text field

- 1 Select Window > Other Panels > Strings.
- 2 Select the Text tool. On the Stage, create an input or dynamic text field.
- 3 While the text field is selected, type a unique ID in the ID field in the Strings panel.
- 4 Click the Settings button and select a language or languages from the list in the Settings dialog box. The languages you select should include the default language you wish to use and any other languages in which you plan to publish your work.
- 5 Click Apply.

Note: If a static text field is selected on the Stage, the Stage text selection section on the Strings panel displays the message “Static text cannot have an ID associated with it.” If a nontext item is selected or multiple items are selected, the message “Current selection cannot have an ID associated with it” appears.

Add a string ID to the Strings panel without assigning it to a text field

- 1 Select Window > Other Panels > Strings.
- 2 Click the Settings button and select a language or languages from the list in the Settings dialog box. The languages you select should include the default language you wish to use and any other languages in which you plan to publish your work.
- 3 Type a new string ID and new string in the Strings panel, and click Apply.

Assign an existing ID to a text field

- 1 Select the Text tool. On the Stage, create an input or dynamic text field.
- 2 Type the name of an existing ID in the ID section of the Strings panel, and click Apply.

Note: Press Shift+Enter to apply the ID to the text field, or Enter if the focus is on the ID field.

Editing strings in the Strings panel

After you enter text strings in the Strings panel, use one of the following methods to edit the text strings:

- Directly in the Strings panel cells.
- On the Stage in the language selected as the Stage language, using features such as find and replace and spelling checking. Text that you change using these features is changed on the Stage and in the Strings panel.
- Edit the XML file directly.

See also

Translate text in the Strings panel or an XML file

[“Find and Replace”](#) on page 58

[“Check spelling”](#) on page 250

Change the language displayed on the Stage

- 1 Select Window > Other Panels > Strings.
- 2 In the Stage Language menu, select the language to use for the Stage language. This must be a language you added as an available language.

After you change the Stage language, any new text you type on the Stage appears in that language. If you previously entered text strings for the language in the Strings panel, any text on the Stage appears in the selected language. If not, the text fields already on the Stage are blank.

Enter Asian characters on a Western keyboard

With Flash, you can use Input Method Editors (IMEs) and standard Western keyboards to enter Asian characters on the Stage. Flash supports more than two dozen IMEs.

For example, to create a website that reaches a broad range of Asian viewers, you can use a standard Western (QWERTY) keyboard and change the IME to create text in Chinese, Japanese, and Korean.

Note: This feature affects only text input on the Stage, not text entered in the Actions panel. This feature is available for all supported Windows operating systems and Mac OS X.

- 1 Select Edit > Preferences (Windows) or Flash > Preferences (Macintosh), and click Text in the Category list.

- 2 Under Input Method, select one of the options to input characters from a Western keyboard. The default is Chinese and Japanese and it should also be selected for Western languages.

Publishing multilanguage FLA files

When you save, publish, or test the FLA file, a folder with an XML file is created for each available language you selected in the Strings panel. The default location for the XML folders and files is the same folder indicated as the SWF publish path. If no SWF publish path was selected, the XML folder and files are saved in the folder in which the FLA file is located. For example, if you have a file named Test in the mystuff folder, and you selected English (en), German (de), and Spanish (es) as active languages, and you did not select a SWF publish path, when you save the FLA file, the following folder structure is created:

```
\mystuff\Test.fl  
a  
\mystuff\de\Test_de.xml  
\mystuff\en\Test_en.xml  
\mystuff\es\Test_es.xml
```

When you start a SWF file, you also need to start the associated XML files with the string translations in the web server. The first frame that contains text cannot appear until the entire XML file is downloaded.

See also

[“Publishing Flash documents”](#) on page 379

Manually replace strings at publish time

Manually replace strings by using the Stage language when you publish your Flash SWF file. This method uses the Stage language to replace all instances of input and dynamic text with an associated string ID. In this case, text strings are only updated when you publish the SWF file; language detection is not automatic, and you must publish a SWF file for each language to support.

- 1 Select Window > Other Panels > Strings, and click Settings.
- 2 Select the Replace Strings Automatically At Runtime check box.

Use automatic language detection with the default language

You can change the default runtime language to any language that you selected as an available language. When automatic language detection is on, and you view the SWF file on the system that uses the language, any system that is set to a language other than one of the active languages uses the default language. For example, if you set your default language to English and you select ja, en, and fr as active languages, users who have their system language set to Japanese, English, or French automatically see text strings in their chosen language. However, users who have their system language set to Swedish, which is not one of the selected languages, automatically see text strings in the default language you selected—in this case, English.

- 1 Select Window > Other Panels > Strings, and click Settings.
- 2 In the Default language menu, select the default language. This must be a language you added as an available language.
- 3 To enable automatic language detection, select Replace Strings Automatically At Runtime, and click OK.

Flash generates the following ActionScript®, which stores the language XML file paths. Use this code as a starting point for your own language detection script.

```
import mx.lang.Locale;
Locale.setFlaName("<flaFileName>");
Locale.setDefaultLang("langcode");
Locale.addXMLPath("langcode", "url/langcode/flaname_langcode.xml");
```

Note: The ActionScript code that the Strings panel generates does not use the `Locale.initialize` function. Decide how to call this function based on the language detection customizations your project requires.

Use custom language detection

To access the language XML files to control text replacement at a time that you designate, create your own custom component or use ActionScript code. For example, you might create a pop-up menu that lets users select a language for viewing content.

For information on writing ActionScript code to create custom language detection, see About the Strings panel in [Learning ActionScript 2.0 in Adobe Flash](#).

- 1 Select Window > Other Panels > Strings, and click Settings.
- 2 In the Default Language menu, select the default language.

This must be a language you added as an available language.

- 3 Select the Replace Strings Via ActionScript check box, and click OK.

Flash generates the following ActionScript code, which stores the language XML file paths. Use this code as a starting point for your own language detection script.

```
import mx.lang.Locale;
Locale.setFlaName("<flaFileName>");
Locale.setDefaultLang("langcode");
Locale.addXMLPath("langcode", "url/langcode/flaname_langcode.xml");
```

Note: The ActionScript that the Strings panel generates does not use the `Locale.initialize` function. Decide how to call this function based on the language detection customizations your project requires.

XML file format for multilanguage text

When you use multilanguage text in Flash, the text is stored in XML files.

About the XML file format

Exported XML is in UTF-8 format and follows the XML Localization Interchange File Format (XLIFF) 1.0 standard. It defines a specification for an extensible localization interchange format that lets any software provider produce a single interchange format that can be delivered to, and understood by, any localization service provider. For more information about XLIFF, see www.oasis-open.org/committees/xliff/.

XLIFF examples

If any of the following characters are entered in the Strings panel, they are replaced by the appropriate entity reference when written to XML files:

Character	Replaced by
&	&
'	'
"	"
<	<
>	>

Exported XML file sample

The following examples show what an XML file that the Strings panel generates looks like in the source language—in this example, English—and in another language—in this example, French:

English source version sample:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE xliiff PUBLIC "-//XLIFF//DTD XLIFF//EN"
"http://www.oasis-open.org/committees/xliiff/documents/xliiff.dtd" >
<xliiff version="1.0" xml:lang="en">
<file datatype="plaintext" original="MultiLingualContent fla" source-language="EN">
  <header></header>
  <body>
    <trans-unit id="001" resname="IDS_GREETINGS">
      <source>welcome to our web site!</source>
    </trans-unit>
    <trans-unit id="002" resname="IDS_MAILING LIST">
      <source>Would you like to be on our mailing list?</source>
    </trans-unit>
    <trans-unit id="003" resname="IDS_SEE YOU">
      <source>see you soon!</source>
    </trans-unit>
    <trans-unit id="004" resname="IDS_TEST">
      <source></source>
    </trans-unit>
  </body>
</file>
</xliiff>
```

French source version sample:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE xliiff PUBLIC "-//XLIFF//DTD XLIFF//EN"
"http://www.oasis-open.org/committees/xliiff/documents/xliiff.dtd" >
<xliiff version="1.0" xml:lang="fr">
<file datatype="plaintext" original="MultiLingualContent fla" source-language="EN">
  <header></header>
  <body>
    <trans-unit id="001" resname="IDS_GREETINGS">
      <source>Bienvenue sur notre site web!</source>
    </trans-unit>
    <trans-unit id="002" resname="IDS_MAILING LIST">
      <source>Voudriez-vous être sur notre liste de diffusion?</source>
    </trans-unit>
    <trans-unit id="003" resname="IDS_SEE YOU">
      <source>A bientôt!</source>
    </trans-unit>
    <trans-unit id="004" resname="IDS_TEST">
      <source></source>
    </trans-unit>
  </body>
</file>
</xliiff>

```

Translate text in the Strings panel or an XML file

When sending files to translators, include not only the FLA file but also the folders for the XML files and the XML file for each language.

Translators can either work directly in the language columns in the Strings panel or work in the XML files for each language to translate the FLA file to selected languages. If you translate directly in the XML file, you must either import the XML file to the Strings panel or save it in the default directory for that language.

Translate text in the Strings panel

- 1 Select Window > Other Panels > Strings.
- 2 For each language to be translated, select the appropriate language column, then type the translated text for that language to be associated with each string ID.
- 3 To show the text on the Stage in the language you selected, select the language in the Stage Language field.
- 4 When you are finished, save, publish, or test the file.

All XML files for all languages are overwritten with the information in the Strings panel.

Note: To preserve the translation in an XML file, save it in a different folder.

Translate text in an XML file

- 1 Using an XML file editor or translating software, open the folder for the desired language, then the XML file for that language. The XML file is populated with the IDs for each text string.
- 2 Enter the text string for the language next to the ID.
- 3 If necessary, import the translated XML file into the Strings panel.

Import an XML file into the Strings panel

After you modify an XML file, if you place it in the folder specified in the Strings panel for that language, the XML file is loaded into the Flash document (FLA file) when it opens.

Regardless of where the XML file you imported was located, when you save, test, or publish the FLA file, a folder for each language in the Strings panel and an XML file for each language are created in the location indicated for publishing SWF files. If no publish path is indicated, the folder and file are saved in the same folder in which the FLA file is located. The XML files that the Strings panel generates are always populated with the information in the Strings panel.

Alternatively, import an XML file into the Strings panel from another location. After you import it, when you save, test, or publish the file, the XML file in the folder specified for that language is overwritten. You cannot import an XML file for a language unless it is already selected as an available language in the Strings panel. You can also add a language and import an XML file with the translation for that language.

- 1 Select Window > Other Panels > Strings, and click Import XML.
- 2 In the Select a Language menu, select the language of the XML file you are importing, and click OK.
- 3 Navigate to the folder and XML file to import.

The XML information is loaded into the column in the Strings panel for the language you selected in step 3.

Note: Select the same language in steps 2 and 3. Otherwise, you could, for example, import a French XML file into the column for German.

See also

[“Publishing overview”](#) on page 379

Multilanguage text and ActionScript

You can control multilanguage text and import multilanguage XML files with ActionScript®.

Use ActionScript to load external files

To load existing XML data, or use a different format for the XML file, use the `loadVariables` action, the `getURL` action, the `LoadVars` object, or the `XML` object to create a document that contains multilanguage text by placing the text in an external text or XML file and loading the file into the movie clip at runtime.

Save the external file in UTF-8 (recommended), UTF-16BE, or UTF-16LE format, using an application that supports the format. If you are using UTF-16BE or UTF-16LE format, the file must begin with a BOM to identify the encoding format to Flash Player. The following table lists the BOM to include to identify the encoding:

Note: Most text editors that can save files in UTF-16BE or LE automatically add the BOMs to the files.

UTF Format	First Byte	Second Byte
UTF-16BE	0xFE	0xFF
UTF-16LE	0xFF	0xFE

Note: If the external file is an XML file, you cannot use an XML encoding tag to change the file encoding. Save the file in a supported Unicode format.

- 1 In the Flash authoring application, create a dynamic or input text field to show the text in the document.
- 2 In the Property inspector, with the text field selected, assign an instance name to the text field.
- 3 Outside of Flash, create a text or XML file that defines the value for the text field variable.
- 4 Save the XML file in UTF-8 (recommended), UTF-16BE, or UTF-16LE format.
- 5 Use one of the following ActionScript procedures to reference the external file and load it into the dynamic or input text field:
 - Use the `loadVariables` action to load an external file.
 - Use the `getURL` action to load an external file from a specified URL.
 - Use the `LoadVars` object (a predefined client-server object) to load an external text file from a specified URL.
 - Use the `XML` object (a predefined client-server object) to load an external XML file from a specified URL. For more information, see XML in the [ActionScript 2.0 Language Reference](#).

See also

[“Using the XMLConnector component to connect to external XML files”](#) on page 272

[“Text”](#) on page 244

Create multilanguage documents using the #include action

To create a document that contains multiple languages, use the `#include` action.

Use an application that supports UTF-8 encoding, such as Dreamweaver, to save the text file in UTF-8 format.

To identify the file as Unicode to the Flash authoring tool, include the following header as the first line of the file:

```
//!-- UTF8
```

Note: Include a space after the second dash (-).

By default, the Flash authoring application assumes that external files that use the `#include` action are encoded in the traditional code page of the operating system running the authoring tool. Using the `//!-- UTF8` header in a file tells the authoring tool that the external file is encoded as UTF-8.

- 1 In the Flash authoring tool, create a dynamic or input text field to display the text in the document.
- 2 In the Property inspector, with the text field selected, assign an instance name to the text field.
- 3 Outside of Flash, create a text file that defines the value for the text field variable. Add the `//!-- UTF8` header at the beginning of the file.
- 4 Save the text file in UTF-8 format.
- 5 To include the external file in the dynamic or input text field, use the `#include` directive. For more information, see `#include` directive in the [ActionScript 2.0 Language Reference](#).

See also

[“Text”](#) on page 244

Creating multilanguage documents by using text variables

To include Unicode-encoded contents in text variables, use the syntax `\uXXXX`, where `XXXX` is the four-digit hexadecimal code point, or *escape* character, for the Unicode character. The Flash authoring tool supports Unicode escape characters through `\uFFFF`. To find the code points for Unicode characters, see the Unicode Standard at Unicode.org.

You can use Unicode escape characters only in text field variables. You cannot include Unicode escape characters in external text or XML files; Flash Player 6 does not recognize Unicode escape characters in external files.

For example, to set a dynamic text field (with the `myTextVar` instance name) that contains Japanese, Korean, Chinese, English, and Greek characters and the Euro sign, enter the following:

```
myTextVar.text = "\u304B\uD55C\u6C49hello\u03BB\u20AC";
```

When the SWF file plays, the following characters appear in the text field:

か한汉helloλ€

For best results when creating a text field that contains multiple languages, use a font that includes all the glyphs your text needs.

See also

[“Create multilanguage documents using the #include action”](#) on page 271

Using the XMLConnector component to connect to external XML files

Use the version 2 XMLConnector component to connect to an external XML document to bind to properties in the document. Its purpose is to read or write XML documents by using HTTP `GET` operations, `POST` operations, or both. It acts as a connector between other components and external XML documents. The XMLConnector communicates with components in your application by using either data-binding features in the Flash authoring environment or ActionScript code. For more information, see XML Connector component in the [ActionScript 2.0 Components Language Reference](#).

Chapter 10: Sound

You can use sound in Adobe® Flash® CS4 Professional in several different ways to make your work more interesting and involving. You can import sounds and edit them after they are imported. You can attach sounds to different kinds of objects and trigger them in different ways, depending on your desired effect.

Using sounds in Flash

About sounds and Flash

Adobe® Flash® CS4 Professional offers several ways to use sound. Make sounds that play continuously, independent of the Timeline, or use the Timeline to synchronize animation to a sound track. Add sounds to buttons to make them more interactive, and make sounds fade in and out for a more polished sound track.

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

If you're creating Flash content for mobile devices, Flash also lets you include device sounds in your published SWF file. Device sounds are encoded in the device's natively supported audio format, such as MIDI, MFi, or SMAF.

You can use shared libraries to link a sound to multiple documents. You can also use the ActionScript® 2.0 `onSoundComplete` event or ActionScript® 3.0 `soundComplete` event to trigger an event based on the completion of a sound.

You can load sounds and control sound playback using prewritten behaviors or media components; the latter also provide a controller for stop, pause, rewind, and so on. You can also use ActionScript 2.0 or 3.0 to load sounds dynamically.

For more information, see `attachSound` (`Sound.attachSound` method) and `loadSound` (`Sound.loadSound` method) in *ActionScript 2.0 Language Reference* or `Sound` class in *ActionScript 3.0 Language and Components Reference*.

See also

[“Sharing library assets”](#) on page 162

Importing sounds

You place sound files into Flash by importing them into the library for the current document.

- 1 Select File > Import > Import To Library.
- 2 In the Import dialog box, locate and open the desired sound file.

Note: You can also drag a sound from a common library into the library for the current document.

Flash stores sounds in the library along with bitmaps and symbols. You need only one copy of a sound file to use that sound multiple ways in your document.

If you want to share sounds among Flash documents, you can include the sounds in shared libraries.

Flash includes a Sounds library containing many useful sounds that can be used for effects. To open the Sounds library, choose Window > Common Libraries > Sounds. To import a sound from the Sounds library to your FLA file, drag the sound from the Sounds library to the Library panel of your FLA file. You can also drag sounds from the Sounds library to other shared libraries.

Sounds can use large amounts of disk space and RAM. However, mp3 sound data is compressed and smaller than WAV or AIFF sound data. Generally, when using WAV or AIFF files, it's best to use 16-22 kHz mono sounds (stereo uses twice as much data as mono), but Flash can import either 8- or 16-bit sounds at sample rates of 11, 22, or 44 kHz. Sounds recorded in formats that are not multiples of 11 kHz (such as 8, 32, or 96 kHz) are resampled when imported into Flash. Flash can convert sounds to lower sample rates on export.

If you want to add effects to sounds in Flash, it's best to import 16-bit sounds. If you have limited RAM, keep your sound clips short or work with 8-bit sounds instead of 16-bit sounds.

See also

[“Work with common libraries”](#) on page 160

Supported sound file formats

You can import the following sound file formats into Flash:

- ASND (Windows or Macintosh). This is the native sound format of Adobe® Soundbooth™.
- WAV (Windows only)
- AIFF (Macintosh only)
- mp3 (Windows or Macintosh)

If you have QuickTime® 4 or later installed on your system, you can import these additional sound file formats:

- AIFF (Windows or Macintosh)
- Sound Designer® II (Macintosh only)
- Sound Only QuickTime Movies (Windows or Macintosh)
- Sun AU (Windows or Macintosh)
- System 7 Sounds (Macintosh only)
- WAV (Windows or Macintosh)

Note: The ASND format is a non-destructive audio file format, native to Adobe Soundbooth. ASND files can contain audio data with effects that can be modified later, Soundbooth multitrack sessions, and snapshots that allow you to revert to a previous state of the ASND file.

Add a sound to the Timeline

You can add a sound to a document using the library, or you can load a sound into a SWF file during runtime, using the `loadSound` method of the Sound object. For more information, see `loadSound` (`Sound.loadSound` method) in the *ActionScript 2.0 Language Reference* or `Sound` class in the *ActionScript 3.0 Language and Components Reference*.

- 1 Import the sound into the library if it has not already been imported.
- 2 Select Insert > Timeline > Layer.

- 3 With the new sound layer selected, drag the sound from the Library panel onto the Stage. The sound is added to the current layer.

You can place multiple sounds on one layer or on layers containing other objects. However, it is recommended that each sound be placed on a separate layer. Each layer acts as a separate sound channel. The sounds on all layers are combined when you play the SWF file.

- 4 In the Timeline, select the first frame that contains the sound file.
- 5 Select Window > Properties, and click the arrow in the lower-right corner to expand the Property inspector.
- 6 In the Property inspector, select the sound file from the Sound pop-up menu.
- 7 Select an effect option from the Effects pop-up menu:

None Applies no effects to the sound file. Select this option to remove previously applied effects.

Left Channel/Right Channel Plays sound in the left or right channel only.

Fade Left To Right/Fade Right To Left Shifts the sound from one channel to the other.

Fade In Gradually increases the volume of a sound over its duration.

Fade Out Gradually decreases the volume of a sound over its duration.

Custom Lets you create custom in and out points of sound using the Edit Envelope.

- 8 Select a synchronization option from the Sync pop-up menu:

Note: If you are placing the sound on a frame other than frame 1 in the main Timeline, select the Stop option.

Event Synchronizes the sound to the occurrence of an event. An event sound, such as a sound that plays when a user clicks a button, plays when its starting keyframe first appears and plays in its entirety, independently of the Timeline, even if the SWF file stops playing. Event sounds are mixed when you play your published SWF file. If an event sound is playing and the sound is instantiated again (for example, by the user clicking the button again), the first instance of the sound continues to play and another instance begins to play simultaneously.

Start The same as Event, except that if the sound is already playing, no new instance of the sound plays.

Stop Silences the specified sound.

Stream Synchronizes the sound for playing on a website. Flash forces animation to keep pace with stream sounds. If Flash can't draw animation frames quickly enough, it skips frames. Unlike event sounds, stream sounds stop if the SWF file stops playing. Also, a stream sound can never play longer than the length of the frames it occupies. Stream sounds are mixed when you publish your SWF file.

An example of a stream sound is the voice of a character in an animation that plays in multiple frames.

Note: If you use an mp3 sound as a stream sound, you must recompress the sound for export. You can export the sound as an mp3 file, with the same compression settings that it had on import.

- 9 Enter a value for Repeat to specify the number of times the sound should loop, or select Loop to repeat the sound continuously.

For continuous play, enter a number large enough to play the sound for an extended duration. For example, to loop a 15-second sound for 15 minutes, enter 60. Looping stream sounds is not recommended. If a stream sound is set to loop, frames are added to the file and the file size is increased by the number of times the sound is looped.

- 10 To test the sound, drag the playhead over the frames containing the sound or use commands in the Controller or the Control menu.

Add a sound to a button

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

- 1 Select the button in the Library panel.
- 2 Select Edit from the Panel menu in the upper-right corner of the panel.
- 3 In the button's Timeline, add a layer for sound (Insert > Timeline > Layer).
- 4 In the sound layer, create a regular or blank keyframe to correspond with the button state to which you want to add a sound (Insert > Timeline > Keyframe or Insert > Timeline > Blank Keyframe).

For example, to add a sound that plays when you click the button, create a keyframe in the frame labeled Down.

- 5 Click the keyframe you created.
- 6 Select Window > Properties.
- 7 In the Property inspector, select a sound file from the Sound pop-up menu.
- 8 Select Event from the Sync pop-up menu.

To associate a different sound with each of the button's keyframes, create a blank keyframe and add another sound file for each keyframe. You can also use the same sound file and apply a different sound effect for each button keyframe.

Synchronize a sound with animation

To synchronize a sound with animation, you start and stop the sound at keyframes.

- 1 Add a sound to a document.
- 2 To synchronize this sound with an event in the scene, select a beginning keyframe that corresponds to the keyframe of the event in the scene. You can select any of the synchronization options.
- 3 Create a keyframe in the sound layer's Timeline at the frame where you want the sound to end. A representation of the sound file appears in the Timeline.
- 4 Select Window > Properties, and click the arrow in the lower-right corner to expand the Property inspector.
- 5 In the Property inspector, select the same sound from the Sound pop-up menu.
- 6 Select Stop from the Sync pop-up menu.

When you play the SWF file, the sound stops playing when it reaches the ending keyframe.

- 7 To play back the sound, simply move the playhead.

Edit a sound in Flash

In Flash, you can define the starting point of a sound or control the volume of the sound as it plays. You can also change the point at which a sound starts and stops playing. This is useful for making sound files smaller by removing unused sections.

- 1 Add a sound to a frame, or select a frame that already contains a sound.
- 2 Select Window > Properties.
- 3 Click the Edit button on the right side of the Property inspector.
- 4 Do any of the following:
 - To change the start and end points of a sound, drag the Time In and Time Out controls in the Edit Envelope.

- To change the sound envelope, drag the envelope handles to change levels at different points in the sound. Envelope lines show the volume of the sound as it plays. To create additional envelope handles (up to eight total), click the envelope lines. To remove an envelope handle, drag it out of the window.
 - To display more or less of the sound in the window, click the Zoom In or Out buttons.
 - To switch the time units between seconds and frames, click the Seconds and Frames buttons.
- 5 To hear the edited sound, click the Play button.

Edit a sound in Soundbooth

If you have Adobe Soundbooth installed, you can use Soundbooth to edit sounds you have imported into your FLA file. After making changes in Soundbooth, when you save the file and overwrite the original, the changes are automatically reflected in the FLA file.

If you change the filename or format of the sound after editing it, you will need to re-import it into Flash.

For a video tutorial about using Flash together with Soundbooth, see Working with Soundbooth and Flash at www.adobe.com/go/lrvid4100_xp.

Note: Soundbooth is available only on Windows computers and Intel®-based Macintoshes.

To edit an imported sound in Soundbooth:

- 1 Right-click (Windows) or Ctrl-click (Macintosh) the sound in the Library panel.
 - 2 Choose Edit in Soundbooth from the context menu. The file opens in Soundbooth.
 - 3 Edit the file in Soundbooth.
 - 4 When you are finished, save the file. To save the changes in a non-destructive format, choose the ASND format.
- If you save the file in a different format from the original, you will need to re-import the sound file into Flash.
- 5 Return to Flash to see the edited version of the sound file in the Library panel.

Note: You cannot edit sounds from the Sounds library (Window > Common Libraries > Sounds) with the Edit in Soundbooth command. To edit these sounds in Soundbooth, open Soundbooth and select the sound from the Resource Central panel. Edit the sound and then import it into Flash.

Using sounds in Flash Lite

Adobe® Flash® Lite supports two types of sound: standard Flash sounds, like those used in Flash desktop applications, and device sounds. Flash Lite 1.0 supports device sounds only; Flash Lite 1.1 and 2.x support both standard sounds and device sounds.

Device sounds are stored in the published SWF file in their native audio format (such as MIDI or MFi); during playback, Flash Lite passes the sound data to the device, which decodes and plays the sound. Because you can't import most device audio formats into Flash, you instead import a *proxy* sound in a supported format (such as mp3 or AIFF) that is replaced with an external device sound that you specify.

You can use device sounds only as event sounds—you can't synchronize device sounds with the Timeline as you can with standard sounds.

Flash Lite 1.0 and Flash Lite 1.1 do not support the following features available in the desktop version of Flash® Player:

- The ActionScript Sound object
- Loading of external mp3 files
- The Speech Audio Compression option

For more information, see “Working with Sound, Video, and Images” in *Developing Flash Lite 2.x Applications* or “Working with Sound” in *Developing Flash Lite 1.x Applications*.

Exporting Sounds

About compressing sounds for export

You can select compression options for individual event sounds and export the sounds with those settings. You can also select compression options for individual stream sounds. However, all stream sounds in a document are exported as a single stream file, using the highest setting of all those applied to individual stream sounds. This includes stream sounds in video objects.

If you select global compression settings for event sounds or stream sounds in the Publish Settings dialog box, these settings are applied to individual event sounds or all stream sounds if you do not select compression settings for the sounds in the Sound Properties dialog box.

You can also override export settings specified in the Sound Properties dialog box by selecting Override Sound Settings in the Publish Settings dialog box. This option is useful if you want to create a larger high-fidelity audio file for local use and a smaller low-fidelity version for the web.

The sampling rate and degree of compression make a significant difference in the quality and size of sounds in exported SWF files. The more you compress a sound and the lower the sampling rate, the smaller the size and the lower the quality. You should experiment to find the optimal balance between sound quality and file size.

When working with imported mp3 files, you can export the files in mp3 format using the same settings that the files had when imported.

Note: In Windows, you can also export all the sounds from a document as a WAV file using *File > Export > Export Movie*.

See also

[“Publishing overview”](#) on page 379

[“Specify publish settings for SWF files”](#) on page 383

[“About exporting from Flash”](#) on page 410

Compress a sound for export

- 1 Do one of the following:
 - Double-click the sound’s icon in the Library panel.
 - Right-click (Windows) or Control-click (Macintosh) a sound file in the Library panel and select Properties from the context menu.
 - Select a sound in the Library panel and select Properties from the Panel menu in the upper-right corner of the panel.
 - Select a sound in the Library panel and click the Properties button at the bottom of the Library panel.
- 2 If the sound file has been edited externally, click Update.
- 3 For Compression, select Default, ADPCM, mp3, Raw, or Speech.

The Default compression option uses the global compression settings in the Publish Settings dialog box when you export your SWF file. If you select Default, no additional export settings are available.

- 4 Set export settings.
- 5 Click Test to play the sound once. Click Stop if you want to stop testing the sound before it finishes playing.
- 6 Adjust export settings if necessary until the desired sound quality is achieved, and then click OK.

ADPCM and Raw compression options

ADPCM compression sets compression for 8- or 16-bit sound data. Use the ADPCM setting when you export short event sounds such as button clicks.

Raw compression exports sounds with no sound compression.

Preprocessing Converts mixed stereo sounds to monaural (mono) when you select Convert Stereo To Mono (mono sounds are unaffected by this option).

Sample Rate Controls sound fidelity and file size. Lower rates decrease file size but can also degrade sound quality. Rate options are as follows:

- **5 kHz** Barely acceptable for speech.
- **11 kHz** The lowest recommended quality for a short segment of music and one-quarter the standard CD rate.
- **22 kHz** A popular choice for web playback and half the standard CD rate.
- **44 kHz** The standard CD audio rate.

Note: Flash cannot increase the kHz rate of an imported sound above the rate at which it was imported.

ADPCM Bits (ADPCM only) Specifies the bit depth of the sound compression. Higher bit depths produce higher quality sound.

mp3 compression options

MP3 Compression Lets you export sounds with mp3 compression. Use mp3 when you are exporting longer stream sounds such as music sound tracks.

If you are exporting a file that was imported in mp3 format, you can export the file using the same settings the file had when it was imported.

Use Imported mp3 Quality Default setting. Deselect to select other mp3 compression settings. Select to export an imported mp3 file with the same settings the file had when it was imported.

Bit Rate Determines the bits per second in the exported sound file. Flash supports 8 through 160 Kbps CBR (constant bit rate). When you export music, set the bit rate to 16 Kbps or higher for best results.

Preprocessing Converts mixed stereo sounds to monaural (mono sounds are unaffected by this option).

Note: The Preprocessing option is available only if you select a bit rate of 20 Kbps or higher.

Quality Determines the compression speed and sound quality:

- **Fast** Yields faster compression but lower sound quality.
- **Medium** Yields somewhat slower compression but higher sound quality.
- **Best** Yields the slowest compression and the highest sound quality.

Speech compression option

Speech compression exports sounds using a compression that is adapted to speech.

Note: Flash Lite 1.0 and Flash Lite 1.1 do not support the Speech compression option. For content targeting those player versions, use mp3, ADPCM, or Raw compression.

Sample rate Controls sound fidelity and file size. A lower rate decreases file size but can also degrade sound quality. Select from the following options:

- **5 kHz** Acceptable for speech.
- **11 kHz** Recommended for speech.
- **22 kHz** Acceptable for most types of music on the web.
- **44 kHz** The standard CD audio rate. However, because compression is applied, the sound is not CD quality in the SWF file.

Guidelines for exporting sound in Flash documents

Besides sampling rate and compression, there are several ways to use sound efficiently in a document and keep file size small:

- Set the in and out points to prevent silent areas from being stored in the Flash file and to reduce the size of the sound data in the file.
- Get more out of the same sounds by applying different effects for sounds (such as volume envelopes, looping, and in/out points) at different keyframes. You can get a number of sound effects by using only one sound file.
- Loop short sounds for background music.
- Do not set streaming sound to loop.
- When exporting audio in embedded video clips, remember that the audio is exported using the global streaming settings selected in the Publish Settings dialog box.
- Use stream synchronization to keep the animation synchronized to your sound track when you preview your animation in the editor. If your computer is not fast enough to draw the animation frames so that they keep up with your sound track, Flash skips frames.
- When exporting QuickTime movies, use as many sounds and channels as you want without worrying about file size. The sounds are combined into a single sound track when you export as a QuickTime file. The number of sounds you use has no effect on the final file size.

Sound and ActionScript

With ActionScript®, you can control sounds at runtime. Using ActionScript allows you to create interaction and other capabilities in your FLA files that is not possible with the Timeline alone.

Control sounds using behaviors

Using sound behaviors, prewritten ActionScript 2.0, you can add sounds to your document and control sound playback. Adding a sound using these behaviors creates an instance of the sound, which is then used to control the sound.

Note: ActionScript 3.0 and Flash Lite 1.x and Flash Lite 2.x do not support behaviors.

Load a sound to a file using a behavior

- 1 Select the object, such as a button, that you want to use to trigger the behavior.

- 2 In the Behaviors panel (Window > Behaviors), click the Add (+) button and select Sound > Load Sound from Library or Sound > Load Streaming mp3 File.
- 3 In the Load Sound dialog box, enter the linkage identifier for a sound from the Library, or the sound location for a streaming mp3 file. Next, enter a name for this instance of the sound, and click OK.
- 4 In the Behaviors panel under event, click On Release (the default event), and select a mouse event from the menu. If you want to use the `onRelease` event, do not change the option.

Play or stop sounds using a behavior

- 1 Select the object, such as a button, that you want to use to trigger the behavior.
- 2 In the Behaviors panel (Window > Behaviors), click the Add (+) button.
- 3 Select Sound > Play Sound, Sound > Stop Sound, or Sound > Stop All Sounds.
- 4 In the dialog box that appears, do one of the following:
 - Enter the linkage identifier and the instance name of the sound you want to play or stop, and click OK.
 - Click OK to verify that you want to stop all sounds.
- 5 In the Behaviors panel under Event, click On Release (the default event) and select a mouse event from the menu. If you want to use the `onRelease` event, do not change the option.

Control sounds with the ActionScript 2.0 Sound object

Use the Sound object in ActionScript 2.0 to add sounds to a document and to control sound objects in a document, including adjusting the volume or the right and left balance while a sound plays. For more information, see [Creating sound controls in *Learning ActionScript 2.0 in Flash*](#).

- 1 Select the sound in the Library panel.
- 2 Select Linkage from the Panel menu in the upper-right corner of the panel, or right-click (Windows) or Control-click (Macintosh) the sound name in the Library panel and select Linkage from the context menu.
- 3 Under Linkage in the Linkage Properties dialog box, select Export for ActionScript.
- 4 Enter an identifier string in the box, and click OK.

About the ActionScript 2.0 onSoundComplete event

The `onSoundComplete` event of the ActionScript 2.0 Sound object lets you trigger an event in a Flash application based on completing an attached sound file. The Sound object is a built-in object that lets you control sounds in a Flash application. For more information, see Sound in the [ActionScript 2.0 Language Reference](#). The `onSoundComplete` event of a Sound object is invoked automatically when the attached sound file finishes playing. If the sound is looped a specified number of times, the event is triggered when the sound finishes looping.

The Sound object has two properties that you can use with the `onSoundComplete` event. The `duration` property is a read-only property representing the duration, in milliseconds, of the sound sample attached to the sound object. The `position` property is a read-only property representing the number of milliseconds the sound has been playing in each loop.

The `onSoundComplete` event lets you manipulate sounds in a many ways, such as the following:

- Creating a dynamic playlist or sequencer
- Creating a multimedia presentation that checks for narration completion before advancing to the next frame or scene

- Building a game that synchronizes sounds to particular events or scenes and transitions smoothly between different sounds
- Timing an image change to a sound—for example, changing an image when a sound is halfway through at playback time

Accessing ID3 properties in mp3 files with Flash Player

Macromedia Flash Player 7 from Adobe and later supports ID3 v2.4 and v2.4 tags. With this version, when you load an mp3 sound using the ActionScript 2.0 `attachSound()` or `loadSound()` method, the ID3 tag properties are available at the beginning of the sound data stream. The `onID3` event executes when the ID3 data is initialized.

Flash Player 6 (6.0.40.0) and later supports mp3 files with ID3 v1.0 and v1.1 tags. With ID3 v1.0 and v1.1 tags, the properties are available at the end of the data stream. If a sound does not contain an ID3v1 tag, the ID3 properties are undefined. Users must have Flash Player 6 (6.0.40.0) or later for the ID3 properties to function.

For more information on using the ID3 properties, see `id3` (`Sound.id3` property) in the [ActionScript 2.0 Language Reference](#).

Chapter 11: Video

Digital video and Flash

Adobe® Flash® CS4 Professional is a powerful tool for incorporating video footage into web-based presentations. The FLV and F4V (H.264) video format offers technological and creative benefits that let you fuse video together with data, graphics, sound, and interactive control. FLV or F4V video lets you easily put video on a web page in a format that almost anyone can view.

How you choose to deploy your video determines how you create your video content, and how you integrate it with Flash. You can incorporate video into Flash in the following ways:

Stream video with Adobe Flash Media Server You can host video content on Adobe® Flash® Media Server, a server solution optimized to deliver real-time media. Flash Media Server uses the Real-Time Messaging Protocol (RTMP), a protocol designed for real-time server applications such as streaming video and audio content. You can host your own Flash Media Server, or use a hosted Flash® Video® Streaming Service (FVSS). Adobe has partnered with several content delivery network (CDN) providers to offer hosted services for delivering on-demand FLV or F4V file video across high-performance, reliable networks. Built with Flash Media Server and integrated directly into the delivery, tracking, and reporting infrastructure of the CDN network, FVSS provides the most effective way to deliver FLV or F4V files to the largest possible audience without the hassle of setting up and maintaining your own streaming server hardware and network.

To create Flash applications using streaming video, you import locally stored video clips into your Flash documents, and later upload them to the server. To control video playback and provide intuitive controls for users to interact with the streaming video, use the FLVPlayback component or Adobe® ActionScript®.

Progressively download video from a web server If you don't have access to Flash Media Server or FVSS, or your video needs are for a low-volume website with only limited amounts of video content, you can consider *progressive downloading*. Progressively downloading a video clip from a web server doesn't provide the real-time performance that Flash Media Server does; however, you can use relatively large video clips, and keep the size of your published SWF files to a minimum.

To control video playback and provide intuitive controls for users to interact with the video, use the FLVPlayback component or ActionScript.

Embed video in the Flash document You can embed a small, short-duration video file directly into the Flash document, and publish it as part of the SWF file. Embedding video content directly into the Flash SWF file significantly increases the size of published file, and is only suitable for small video files (typically less than 10 seconds in length). In addition, the audio to video synchronization (also known as audio/video sync) can become mis-synchronized when using longer video clips embedded in the Flash document. Another disadvantage to embedding video within the SWF file is that you cannot update the video without republishing the SWF file.

For video tutorials about working with video in Flash, see the following:

- Using video in Flash: www.adobe.com/go/vid0136
- Creating FLV and F4V files: www.adobe.com/go/lrvid4097_xp
- Using Adobe Media Encoder: www.adobe.com/go/vid0138
- Creating content for Adobe After Effects: www.adobe.com/go/vid0139
- Working with Flash and After Effects: www.adobe.com/go/lrvid4098_xp

For a text tutorial about using video in Flash, see Building a Video Player on the Flash Tutorials page at www.adobe.com/go/learn_fl_tutorials.

See also

[“Import video for streaming or progressive download”](#) on page 288

Control video playback

You can control the playback of video in Flash using the FLVPlayback component, by writing custom ActionScript to play an external video stream, or by writing custom ActionScript to control the playback of video in the Timeline for embedded video.

FLVPlayback component Lets you quickly add a full-featured FLV playback control to your Flash document and provides support for both progressive downloading and streaming FLV or F4V files. FLVPlayback lets you easily create intuitive video controls for users to control video playback and apply pre-made skins, or apply your own custom skins to the video interface. For more information see [“The FLVPlayback component”](#) on page 298.

Control external video using ActionScript Play back external FLV or F4V files in a Flash document at runtime using the `NetConnection` and `NetStream` ActionScript objects. For more information see [“Controlling external video playback with ActionScript”](#) on page 296.

You can use video behaviors (pre-written ActionScript scripts) to control video playback.

Control embedded video in the Timeline To control playback of embedded video files, you must write ActionScript to control the Timeline containing the video. For more information see [“Control video playback using the Timeline”](#) on page 292.

See also

[“The FLVPlayback component”](#) on page 298

[“Controlling external video playback with ActionScript”](#) on page 296

[“Control video playback using behaviors”](#) on page 298

The Video Import Wizard

The Video Import Wizard simplifies the importing of video into a Flash document by guiding you through the process of selecting an existing video file, and importing the file for use in one of three different video playback scenarios. The Video Import Wizard provides a basic level of configuration for the import and playback method you choose, which you can later modify for your specific requirements.

The Video Import dialog box provides three video import options:

Load external video with playback component Imports the video and creates an instance of the FLVPlayback component to control video playback. When you are ready to publish the Flash document as a SWF and upload it to your web server, you must also upload the video file to either a web server or Flash Media Server, and configure the FLVPlayback component with the location of the uploaded video file.

Embed FLV or F4V in SWF and play in timeline Embeds the FLV or F4V into the Flash document. When you import video this way, the video is placed in the Timeline where you can see the individual video frames represented in the Timeline frames. An embedded FLV or F4V video file becomes part of the Flash document.

***Note:** Embedding video content directly into the Flash SWF file significantly increases the size of published file, and is only suitable for small video files. In addition, the audio to video synchronization (also known as audio/video sync) can become mis-synchronized when using longer video clips embedded in the Flash document.*

Import as mobile device video bundled in SWF Similar to embedding a video in a Flash document, you bundle a video into a Flash Lite document for deployment to a mobile device. For information on using video in Flash Lite documents, see [Working with video](#) in *Developing Flash Lite 2.x and 3.x Applications*.

Video formats and Flash

To import video into Flash you must use video encoded in the FLV or H.264 format. The Video Import Wizard (File > Import > Import Video) checks video files that you select for import, and alerts you if the video might not be in a format that Flash can play. In the event that the video is not in either the FLV or F4V format, you can use Adobe® Media® Encoder to encode the video in the appropriate format.

Adobe Media Encoder

Adobe® Media® Encoder is a stand-alone encoding application employed by programs such as Adobe® Premiere® Pro, Adobe® Soundbooth®, and Flash for output to certain media formats. Depending on the program, the Adobe Media Encoder provides a specialized Export Settings dialog box that accommodates the numerous settings associated with certain export formats, such as Adobe Flash Video and H.264. For each format, the Export Settings dialog box includes a number of presets that are tailored for particular delivery media. You can also save custom presets, which you can share with others or reload as needed.

For information on encoding video in the FLV or F4V format using Adobe Media Encoder, see [About Adobe Media Encoder](#).

The H.264, On2 VP6, and Sorenson Spark video codecs

When encoding video using Adobe Media Encoder, you can choose from three different video codecs with which to encode your video content for use with Flash:

H.264 Support for the H.264 video codec was incorporated into Flash Player beginning with version 9.0.r115. The F4V video format which uses this codec provides a significantly better quality-to-bitrate ratio than previous Flash video codecs, however, it is more computationally demanding than the Sorenson Spark and On2 VP6 video codecs released with Flash Player 7 and 8.

***Note:** If you need to use either video with alpha channel support for compositing, or cue points to trigger synchronized events with video playback, you must use the On2 VP6 video codec; F4V does not support alpha video channels. Cue points can be embedded in the XMP metadata of an F4V file, but parsing that data requires custom ActionScript code.*

On2 VP6 The On2 VP6 codec is the preferred video codec to use when creating FLV files you intend to use with Flash Player 8 and higher. The On2 VP6 codec provides:

- Higher quality video when compared to the Sorenson Spark codec encoded at the same data rate
- Support for the use of an 8-bit alpha channel to composite video

To support better quality video at the same data rate, the On2 VP6 codec is noticeably slower to encode and requires more processor power on the client computer to decode and play back. For this reason, carefully consider the lowest common denominator of computer you intend your viewing audience to use when accessing your FLV video content.

Sorenson Spark Introduced in Flash Player 6, the Sorenson Spark video codec should be used if you intend to publish Flash documents requiring backwards compatibility to Flash Player 6 and 7. If you anticipate a large user base that uses older computers, you should consider FLV files encoded with the Sorenson Spark codec, as it is much less computationally demanding to play back than either the On2 VP6 or H.264 codecs.

If your Flash content dynamically loads Flash video (using either progressive download or Flash Media Server), you can use On2 VP6 video without having to republish a SWF file originally created for use with Flash Player 6 or 7, as long as users use Flash Player 8 or later to view your content. By streaming or downloading On2 VP6 video into Flash SWF versions 6 or 7, and playing the content using Flash Player 8 or later, you avoid having to recreate your SWF files for use with Flash Player 8 and later versions.

Important: Only Flash Player 8 and 9 support both publish and playback of On2 VP6 video.

Codec	SWF version (publish version)	Flash Player version (required for playback)
Sorenson Spark	6	6, 7, 8
	7	7, 8, 9, 10
On2 VP6	6, 7, 8	8, 9, 10
H.264	9.2 or later	9.2 or later

Tips for creating Adobe FLV and F4V video

Follow these guidelines to deliver the best possible FLV or F4V video:

Work with video in the native format of your project until your final output

If you convert a precompressed digital video format into another format such as FLV or F4V, the previous encoder can introduce video noise. The first compressor already applied its encoding algorithm to the video, reducing its quality, frame size, and rate. That compression may have also introduced digital artifacts or noise. This additional noise affects the final encoding process, and a higher data rate may be required to encode a good-quality file.

Strive for simplicity

Avoid elaborate transitions—they don’t compress well and can make your final compressed video look “chunky” during the change. Hard cuts (as opposed to dissolves) are usually best. Eye-catching video sequences—for instance showing an object zooming from behind the first track, doing a “page peel,” or wrapping around a ball and then flying off the screen—don’t compress well and should be used sparingly.

Know your audience data rate

When you deliver video over the Internet, produce files at lower data rates. Users with fast Internet connections can view the files with little or no delay for loading, but dial-up users must wait for files to download. Make the clips short to keep the download times within acceptable limits for dial-up users.

Select the proper frame rate

Frame rate indicates frames per second (fps). If you have a higher data rate clip, a lower frame rate can improve playback through limited bandwidth. For example, if you are compressing a clip with little motion, cutting the frame rate in half probably saves you only 20% of the data rate. However, if you are compressing high-motion video, reducing the frame rate has a much greater effect on the data rate.

Because video looks much better at native frame rates, leave the frame rate high if your delivery channels and playback platforms allow. For web delivery, get this detail from your hosting service. For mobile devices, use the device-specific encoding presets, and the device emulator available through Adobe Media Encoder in Adobe Premiere Pro. If you need to reduce the frame rate, the best results come from dividing the frame rate by whole numbers.

Select a frame size that fits your data rate and frame aspect ratio

At a given data rate (connection speed), increasing the frame size decreases video quality. When you select the frame size for your encoding settings, consider frame rate, source material, and personal preferences. To prevent pillarboxing, it's important to choose a frame size of the same aspect ratio as that of your source footage. For example, you get pillarboxing if you encode NTSC footage to a PAL frame size.

Adobe Media Encoder makes several Adobe FLV or F4V video presets available. These include preset frame sizes and frame rates for the different television standards at different data rates. Use the following list of common frame sizes (in pixels) as a guide, or experiment with the various Adobe Media Encoder presets to find the best setting for your project.

Dial-up Modem NTSC 4 x 3 162 x 120

Dial-up Modem PAL 4 x 3 160 x 120

T1/DSL/cable NTSC 4 x 3 648 x 480

T1/DSL/cable PAL 4 x 3 768 x 576

Stream for best performance

To eliminate download time, provide deep interactivity and navigation capabilities, or monitor quality of service, stream Adobe FLV or F4V video files with the Flash Media Server or use the hosted service from one of Adobe's Flash Video Streaming Service partners available through the Adobe website. For more details on the difference between Progressive Download and Streaming with Flash Media Server, see "Delivering Flash Video: Understanding the Difference Between Progressive Download and Streaming Video" on the Flash Developer Center website.

Know progressive download times

Know how long it will take to download enough of your video so that it can play to the end without pausing to finish downloading. While the first part of your video clip downloads, you may want to display other content that disguises the download. For short clips, use the following formula: $\text{Pause} = \text{download time} - \text{play time} + 10\% \text{ of play time}$. For example, if your clip is 30 seconds long and it takes one minute to download, give your clip a 33-second buffer ($60 \text{ seconds} - 30 \text{ seconds} + 3 \text{ seconds} = 33 \text{ seconds}$).

Remove noise and interlacing

For the best encoding, you might need to remove noise and interlacing.

The higher the quality of the original, the better the final result. Although frame rates and sizes of Internet video are usually smaller than those of television, computer monitors have much better color fidelity, saturation, sharpness, and resolution than conventional televisions. Even with a small window, image quality can be more important for digital video than for standard analog television. Artifacts and noise that are barely noticeable on TV can be obvious on a computer screen.

Adobe Flash is intended for progressive display on computer screens and other devices, rather than on interlaced displays such as TVs. Interlaced footage viewed on a progressive display can exhibit alternating vertical lines in high-motion areas. Thus, Adobe Media Encoder removes interlacing from all video footage that it processes.

Follow the same guidelines for audio

The same considerations apply to audio production as to video production. To achieve good audio compression, begin with clean audio. If you are encoding material from a CD, try to record the file using direct digital transfer instead of through the analog input of your sound card. The sound card introduces an unnecessary digital-to-analog and analog-to-digital conversion that can create noise in your source audio. Direct digital transfer tools are available for Windows and Macintosh platforms. To record from an analog source, use the highest-quality sound card available.

Note: If your source audio file is monaural (mono), it is recommended that you encode in mono for use with Flash. If you are encoding with Adobe Media Encoder, and using an encoding preset, be sure to check if the preset encodes in stereo or mono, and select mono if necessary.

See also

[“Video formats and Flash”](#) on page 285

[“The H.264, On2 VP6, and Sorenson Spark video codecs”](#) on page 285

Import video for streaming or progressive download

Flash provides several methods by which you can import video into Flash, incorporate the video into your Flash document, and play it back for users.

Stream video using Flash Media Server

In streaming, each Flash client opens a persistent connection to the Flash Media Server, and a controlled relationship exists between the video being delivered and the client interaction. Flash Media Server uses bandwidth detection to deliver video or audio content based on the user’s available bandwidth. This lets you provide different content for users based on their ability to easily access and download content. For example, if a user with a dial-up modem accesses your video content, you can deliver an appropriately encoded file that doesn’t require too much bandwidth.

Flash Media Server also provides you with quality of service metrics, detailed tracking and reporting statistics, and a range of interactive features designed to enhance the video experience. As with progressive downloading, the video content (FLV or F4V file) is kept external to the other Flash content and the video playback controls. This lets you easily add or change content without having to republish the SWF file.

Streaming video with Flash Media Server or FVSS provides the following advantages over embedded and progressively downloaded video:

- Video playback starts sooner than it does using other methods of incorporating video.
- Streaming uses less of the client’s memory and disk space, because the clients don’t need to download the entire file.
- Network resources are used more efficiently, because only the parts of the video that are viewed are sent to the client.
- Delivery of media is more secure, because media is not saved to the client’s cache when streamed.
- Streaming video provides better tracking, reporting, and logging ability.
- Streaming lets you deliver live video and audio presentations, or capture video from a web cam or digital video camera.
- Flash Media Server enables multiway and multiuser streaming for video chat, video messaging, and video conferencing applications.
- By using server-side scripting to control video and audio streams, you can create server-side play lists, synchronized streams, and more intelligent delivery options based on the client’s connection speed.

To learn more about Flash Media Server, see www.adobe.com/go/flash_media_server.

To learn more about FVSS, see www.adobe.com/go/fvss.

See also

“[The FLVPlayback component](#)” on page 298

Progressively download video using a web server

Progressive downloading lets you use either the FLVPlayback component or ActionScript that you write to load external FLV or F4V files into a SWF file, and play them back at runtime.

Because the video content is kept external to the other Flash content and the video playback controls, it’s relatively easy to update video content without republishing the SWF file.

Progressive downloading provides the following advantages over embedded video:

- During authoring, publish only the SWF file interface to preview or test part or all of your Flash content. This results in faster preview times and quicker turnaround on iterative experimentation.
- During delivery, video begins playing as soon as the first segment is downloaded and cached to the local computer’s disk drive.
- At runtime, video files are loaded from the computer’s disk drive into the SWF file, with no limitation on file size or duration. No audio synchronization issues or memory restrictions exist.
- The frame rate of the video file can be different from the frame rate of the SWF file, allowing for greater flexibility in authoring your rich media content.

Import video for progressive download or streaming

Import a video file that is already deployed to a web server or a Flash Media Server or Flash Video Streaming Service (FVSS), or select a video file that is stored locally on your computer, and upload the video file to the server after importing it to your FLA file.

- 1 Select File > Import > Import Video to import the video clip into the current Flash document.
- 2 Select the video clip to import. You can select either a video clip located on your local computer, or enter the URL of a video already uploaded to a web server or Flash Media Server:
 - To import video located on your local computer, select Load external video with playback component.
 - To import video already deployed to a web server, Flash Media Server, or FVSS, select Already deployed to a web server, Flash Video Streaming Service, or Stream From Flash Media Server, and enter the URL of the video clip.

Note: The URL for a video clip located on a web server will use the http communication protocol. The URL for a video clip located on a Flash Media Server or Flash Streaming Service will use the RTMP communication protocol.

- 3 Select a skin for your video clip. You can choose to:
 - Not use a skin with the FLVPlayback component by selecting None.
 - Select one of the predefined FLVPlayback component skins. Flash copies the skin into the same folder as the FLA file.

Note: FLVPlayback component skins are slightly different depending on whether you are creating an AS2- or AS3-based Flash document.

- Select a custom skin of your own design by entering the URL of the skin on the web server.
- 4 The Video Import Wizard creates an FLVPlayback video component on the Stage that you can use to test video playback locally. When you finish creating your Flash document and want to deploy the SWF file and video clip, upload the following assets to the web server or Flash Media Server hosting your video:
 - If using a local copy of the video clip, upload the video clip (which is located in the same folder as the source video clip you selected with a .flv extension)

Note: Flash uses a relative path to point to the FLV or F4V file (relative to the SWF file), letting you use the same directory structure locally that you use on the server. If the video was previously deployed to your FMS or the FVSS hosting your video, you can skip this step.

- The video skin (if you chose to use a skin)

To use a predefined skin, Flash copies the skin into the same folder as the FLA file.

- The FLVPlayback component

To edit the FLVPlayback component's URL field to that of the web server or Flash Media Server that you are uploading the video to, use the Component inspector (Windows > Component inspector) to edit the `contentPath` parameter.

See also

[“Specify the contentPath or source parameter”](#) on page 299

[“The FLVPlayback component”](#) on page 298

Embed video in a SWF file

When you embed video, the video file becomes part of the Flash document. The video is placed in the Timeline where you can see the individual video frames represented in the Timeline frames. Because each video frame is represented by a frame in the Timeline, the frame rate of the video clip and the SWF file must be the same. If you use different frame rates for the SWF file and the embedded video clip, playback is inconsistent.

Note: To use variable frame rates, stream the video using either progressive downloading or Flash Media Server. When you import video files using either of these methods, the FLV or F4V files are self-contained and run at a frame rate separate from that of all other timeline frame rates included in the SWF file.

Embedded video works best for smaller video clips, with a playback time of less than 10 seconds. If you are using video clips with longer playback times, consider using progressively downloaded video, or streaming video using Flash Media Server.

The limitations of embedded video include:

- You might encounter problems if the resulting SWF files become excessively large. Flash Player reserves a lot of memory when downloading and attempting to play large SWF files with embedded video, which can cause Flash Player to fail.
- Longer video files (over 10 seconds long) often have synchronization issues between the video and audio portions of a video clip. Over time, the audio track begins playing out of sequence with the video, causing a less than desirable viewing experience.
- To play a video embedded in a SWF file, the entire video file must be downloaded before the video starts to play. If you embed an excessively large video file, it might take a long time for the SWF file to download in its entirety and for playback to start.
- After a video clip is imported, it cannot be edited. Instead, you must re-edit and re-import the video file.
- When publishing your SWF file via the web, the entire video must be downloaded to the viewer's computer before video playback can begin.
- At runtime, the entire video must fit into the local memory of the playback computer.
- The length of an imported video file cannot exceed 16000 frames.

- The video frame rate and Flash Timeline frame rate must be the same. Set the frame rate of your Flash file to match the frame rate of the embedded video.

You can preview frames of an embedded video by dragging the playhead along the Timeline (scrubbing). Note that the video sound track does not play back during scrubbing. To preview the video with sound, use the Test Movie command.

See also

[“Video formats and Flash”](#) on page 285

[“Test document download performance”](#) on page 443

[“About symbols”](#) on page 147

Embed video within the SWF file

- 1 Select File > Import > Import Video to import the video clip into the current Flash document.
- 2 Select the video clip on your local computer to import.
- 3 Select Embed FLV In SWF and Play In Timeline.
- 4 Click Next.
- 5 Choose the symbol type with which to embed the video in the SWF file.

Embedded Video If you’re using the video clip for linear playback in the Timeline, importing the video into the Timeline is the most appropriate method.

Movie Clip A best practice is to place video inside a movie clip instance, because you have the most control over the content. The video’s Timeline plays independently from the main Timeline. You do not have to extend your main Timeline by many frames to accommodate the video, which can make working with your FLA file difficult.

Graphic When you embed a video clip as a graphic symbol, you cannot interact with the video using ActionScript (typically you use graphic symbols for static images and to create reusable pieces of animation that are tied to the main Timeline).

- 6 Import the video clip directly onto the Stage (and the Timeline) or as a library item.

By default, Flash places the video you import on the Stage. To import into the library only, deselect Place Instance on Stage.

If you’re creating a simple video presentation with linear narration and little to no interaction, accept the default setting and import the video to the Stage. To create a more dynamic presentation, work with multiple video clips, or add dynamic transitions or other elements using ActionScript, import the video into the library. After a video clip is in the library, customize it by converting it into a MovieClip object that you can more easily control with ActionScript.

By default, Flash expands the Timeline to accommodate the playback length of the video clip you are embedding.

- 7 Click Finish.

The Video Import wizard embeds the video into the SWF file. The video appears either on the Stage or in the library depending on the embedding options you chose.

- 8 In the Property inspector (Window > Properties), give the video clip an instance name, and make any modifications to the video clip’s properties.

Update an embedded video clip after editing it in an external editor

- 1 Select the video clip in the Library Panel.
- 2 Select Properties and click Update.

The embedded video clip is updated with the edited file. The compression settings you selected when you first imported the video are reapplied to the updated clip.

Control video playback using the Timeline

To control playback of an embedded video file, control the Timeline that contains the video. For example, to pause a video playing on the main Timeline, you would call a `stop()` action that targets that Timeline. Similarly, you can control a video object in a movie clip symbol by controlling the playback of that symbol's Timeline.

You can apply the following actions to imported video objects in movie clips: `goto`, `play`, `stop`, `toggleHighQuality`, `stopAllSounds`, `getUrl`, `FSCommand`, `loadMovie`, `unloadMovie`, `ifFrameLoaded`, and `onMouseEvent`. To apply actions to a Video object, first convert the Video object to a movie clip.

To show a live video stream from a camera, use ActionScript. First, place a Video object on the Stage, select New Video from the Library Panel menu. To attach the video stream to the Video object, use `video.attachVideo`.

See also Video and `attachVideo` (`Video.attachVideo` method) in the *ActionScript 2.0 Language Reference*, and `fl.video` in the *ActionScript 3.0 Language Reference*.

See also

[“Playing external FLV or F4V files dynamically”](#) on page 296

Import video files into the library

To import files in the FLV or F4V format, use the Import or Import To Library commands or the Import button in the Video Properties dialog box.

To create your own video player, which dynamically loads FLV or F4V files from an external source, place your video inside a movie clip symbol. When you load FLV or F4V files dynamically, adjust the dimensions of the movie clip to match the actual dimension of the video file and scale the video by scaling the movie clip.

Note: *A best practice is to place video inside a movie clip instance, which gives you the most control over the content. The video's Timeline plays independently from the main Timeline. You do not have to extend your main Timeline by many frames to accommodate the video, which can make working with your FLA file difficult.*

- ❖ To import an FLV or F4V file into the library, do one of the following:
 - Select File > Import > Import To Library.
 - Select any existing video clip in the Library Panel, and select Properties from the Library Panel menu. Click Import. Locate the file to import, and click Open.

Change the properties of a video clip

You can change properties for an instance of an embedded video clip on the Stage, assign the instance an instance name, and change its width, height, and position on the Stage using the Property inspector. You can also *swap* an instance of a video clip—assign a different symbol to an instance of a video clip. Assigning a different symbol to an instance displays a different instance on the Stage but leaves all the other instance properties (such as dimensions and registration point) intact.

In the Video Properties dialog box, you can do the following:

- View information about an imported video clip, including its name, path, creation date, pixel dimensions, length, and file size
- Change the video clip name
- Update the video clip if you modify it in an external editor
- Import an FLV or F4V file to replace the selected clip
- Export a video clip as an FLV or F4V file

For lessons on working with video, see the Adobe Flash Support Center at www.adobe.com/go/flash_video.

Change video instance properties in the Property inspector

- 1 Select an instance of an embedded or linked video clip on the Stage.
- 2 Select Window > Properties, and do any of the following:
 - Enter an instance name in the Name text field on the left side of the Property inspector.
 - Enter values for W and H to change the dimensions of the video instance.
 - Enter values for X and Y to change the position of the upper-left corner of the instance on the Stage.
 - Click Swap. Select a video clip to replace the clip currently assigned to the instance.

Note: You can swap an embedded video clip only with another embedded video clip, and you can swap a linked video clip only with another linked video clip.

View video clip properties in the Video Properties dialog box

- 1 Select a video clip in the Library Panel.
- 2 Select Properties from the Library Panel menu, or click the Properties button located at the bottom of the Library Panel. The Video Properties dialog box is displayed.

Assign a new name to, update, or replace a video with an FLV or F4V file

- 1 Select the video clip in the Library Panel and select Properties from the Library Panel menu.
- 2 Do one of the following:
 - To assign a new name, enter the name in the Name text field.
 - To update a video, navigate to the updated video file and click Open.
 - To replace a video with an FLV or F4V file, click Import, navigate to the FLV or F4V file to replace the current clip, and click Open.

Export QuickTime video files

Using Flash, you can create QuickTime® movies (MOV files) that can be played back by users who have the QuickTime plug-in installed on their computers. This is often done when people use Flash to create title sequences or animation for use as video content. The published QuickTime file can be distributed as a DVD, or incorporated into other applications such as Adobe® Director® or Adobe® Premiere® Pro.

If you are creating a QuickTime video using Flash, set your publish setting to Flash 3, 4, or 5.

Note: The QuickTime Player does not support Flash Player files later than version 5.

See also

[“About QuickTime”](#) on page 416

[“Export QuickTime”](#) on page 416

Working with Adobe Premiere Pro and After Effects

Working with Adobe Premiere Pro and Adobe Flash

Adobe Premiere Pro is a professional tool for editing video. If you use Adobe Flash Professional to design interactive content for websites or mobile devices, you can use Adobe Premiere Pro to edit the movies for those projects. Adobe Premiere Pro gives you professional tools for frame-accurate video editing, including tools for optimizing video files for playback on computer screens and mobile devices.

Adobe Flash Professional is a tool for incorporating video footage into presentations for the web and mobile devices. Adobe Flash offers technological and creative benefits that let you fuse video with data, graphics, sound, and interactive control. The FLV format lets you put video on a web page in a format that almost anyone can view.

If you use Adobe Premiere Pro to export FLV files, you can use Adobe Flash to embed those files into interactive websites or applications for mobile devices. Adobe Flash can import sequence markers you add in an Adobe Premiere Pro sequence as cue points. You can use these cue points to trigger events in SWF files on playback.

If you export video files in other standard formats, Adobe Flash can encode your videos within rich media applications, using the latest compression technologies to deliver the greatest quality possible at small file sizes.

Moving content between Adobe Premiere Pro and Adobe Flash

After you start and edit a video in Adobe Premiere Pro, you can add sequence markers to the timeline. If you select Flash Cue Point for these markers, they can serve as cue points in a rich media application. There are two types of cue point markers: event and navigational cue point markers. You can use navigational cue point markers to navigate to different sections of FLV files, and to trigger the display of on-screen texts. You can use event cue point markers for triggering action scripts at specified times in FLV files.

You can export a movie from Adobe Premiere Pro directly into the FLV format. You can choose from several Export Settings presets that balance file size against audio and video quality to achieve the bit rate needed for any target audience or device. If you export the movie with an alpha channel, you can use the movie easily used as a layer in a Flash project.

You can then import this movie into Adobe Flash for use in an interactive website or mobile application. Flash will read sequence markers as navigational or event cue points. In Flash, you can also customize the interface that surrounds your video.

Alternatively, because Flash can be used to create animations, you can start a movie as a Flash project, export it as an FLV file, then import the FLV file into Adobe Premiere Pro for editing. In Adobe Premiere Pro, for example, you could add titles or mix the animation with other video sources.

Working with Flash and After Effects

If you use Adobe® Flash® to create video or animation, you can use After Effects® to edit and refine the video. For example, from Flash you can export animations and applications as QuickTime movies or Flash Video (FLV) files. You can then use After Effects to edit and refine the video.

If you use After Effects to edit and composite video, you can then use Flash to publish that video. You can also export an After Effects composition as XFL content for further editing in Flash.

For video tutorials about using Flash together with After Effects, see the following:

- “Importing and exporting XFL files between Flash and After Effects” at www.adobe.com/go/lrvid4098_xp.
- “Exporting an After Effects composition to Flash Professional using SWF, F4V/FLV, and XFL” at www.adobe.com/go/lrvid4105_xp.
- “Converting metadata and markers to cue points for use in Flash” at www.adobe.com/go/lrvid4111_xp.

Exporting QuickTime video from Flash

If you create animations or applications with Flash, you can export them as QuickTime movies using the File > Export > Export Movie command in Flash. For a Flash animation, you can optimize the video output for animation. For a Flash application, Flash renders video of the application as it runs, allowing the user to manipulate it. This lets you capture the branches or states of your application that you want to include in the video file.

Rendering and exporting FLV and F4V files from After Effects

When you render finished video from After Effects, select FLV or F4V as the output format to render and export video that can play in Flash Player. You can then import the FLV or F4V file into Flash and publish it in a SWF file, which can be played by Flash Player.

Importing and publishing video in Flash

When you import an FLV or F4V file into Flash, you can use various techniques, such as scripting or Flash components, to control the visual interface that surrounds your video. For example, you might include playback controls or other graphics. You can also add graphic layers on top of the FLV or F4V file for composite results.

Composite graphics, animation, and video

Flash and After Effects each include many capabilities that allow you to perform complex compositing of video and graphics. Which application you choose to use will depend on your personal preferences and the type of final output you want to create.

Flash is the more web-oriented of the two applications, with its small final file size. Flash also allows for run-time control of animation. After Effects is oriented toward video and film production, provides a wide range of visual effects, and is generally used to create video files as final output.

Both applications can be used to create original graphics and animation. Both use a timeline and offer scripting capabilities for controlling animation programmatically. After Effects includes a larger set of effects, while the Flash ActionScript® language is the more robust of the two scripting environments.

Both applications allow you to place graphics on separate layers for compositing. These layers can be turned on and off as needed. Both also allow you to apply effects to the contents of individual layers.

In Flash, composites do not affect the video content directly; they affect only the appearance of the video during playback in Flash Player. In contrast, when you composite with imported video in After Effects, the video file you export actually incorporates the composited graphics and effects.

Because all drawing and painting in After Effects is done on layers separate from any imported video, it is always non-destructive. Flash has both destructive and nondestructive drawing modes.

Exporting After Effects content for use in Flash

You can export After Effects content for use in Flash. You can export a SWF file that can be played immediately in Flash Player or used as part of another rich media project. When you export content from After Effects in SWF format, some of the content may be flattened and rasterized in the SWF file.

To edit your After Effects content further in Flash, export a composition as an XFL file. An XFL file is a type of Flash file that stores the same information as a FLA file, but in XML format. When you export a composition from After Effects as XFL for use in Flash, some of the layers and keyframes that you created in After Effects are preserved in the Flash version. When you import the XFL file in Flash, it unpacks the XFL file and adds the assets from the file to your FLA file according to the instructions in the XFL file.

For a video tutorial about exporting XFL files from After Effects, see “Importing and exporting XFL files between Flash and After Effects” at www.adobe.com/go/lrvid4098_xp.

Importing Flash SWF files into After Effects

Flash has a unique set of vector art tools that make it useful for a variety of drawing tasks not possible in After Effects or Adobe® Illustrator®. You can import SWF files into After Effects to composite them with other video or render them as video with additional creative effects. Interactive content and scripted animation are not retained. Animation defined by keyframes is retained.

Each SWF file imported into After Effects is flattened into a single continuously rasterized layer, with its alpha channel preserved. Continuous rasterization means that graphics stay sharp as they are scaled up. This import method allows you to use the root layer or object of your SWF files as a smoothly rendered element in After Effects, allowing the best capabilities of each tool to work together.

See also

“Digital video and Flash” on page 283

“Opening XFL files” on page 47

Controlling external video playback with ActionScript

With Adobe® ActionScript®, you can play back and control external FLV or F4V video files at runtime. Using ActionScript allows you to create interaction and other capabilities in your FLA files that is not possible with the Timeline alone.

Playing external FLV or F4V files dynamically

An alternative to importing video into the Flash authoring environment is to use either the FLVPlayback component or ActionScript to dynamically play external FLV or F4V files in Flash Player. You can also use the FLVPlayback component and ActionScript together.

You can play FLV or F4V files posted as HTTP downloads or as local media files. To play back an external FLV or F4V file, post an FLV or F4V file to a URL (either an HTTP site or a local folder) and add either the FLVPlayback component or ActionScript code to the Flash document to access the file and control playback during runtime.

Using external FLV or F4V files provides the following capabilities that are not available when using imported video:

- You can use longer video clips without slowing down playback. External FLV or F4V files are played using *cached memory*, which means that large files are stored in small pieces and accessed dynamically; they do not require as much memory as embedded video files.

- An external FLV or F4V file can have a different frame rate from the Flash document in which it plays. For example, you can set the Flash document frame rate to 30 fps and the video frame rate to 21 fps, which gives you greater control in ensuring smooth video playback.
- With external FLV or F4V files, Flash document playback does not have to be interrupted while the video file is loading. Imported video files can sometimes interrupt document playback to perform certain functions (for example, to access a CD-ROM drive). FLV or F4V files can perform functions independently of the Flash document, and so do not interrupt playback.
- Captioning video content is easier with external FLV or F4V files because you can use callback functions to access metadata for the video.

For more information on playing back FLV or F4V files, see “Playing back external FLV files dynamically” in *Learning ActionScript 2.0 in Adobe Flash* or Basics of video in *Programming ActionScript 3.0*.

The Creative Cow web site provides an excellent tutorial on [creating a video player with ActionScript 2.0](#) and another on [creating a video player with ActionScript 3.0](#).

See also

“Stream video using Flash Media Server” on page 288

“Progressively download video using a web server” on page 289

Behaviors used in video playback

Video behaviors provide one way to control video playback. Behaviors are prewritten ActionScript scripts that you add to a triggering object to control another object. Behaviors add the power, control, and flexibility of ActionScript coding to your document without having to create the ActionScript code. Video behaviors play, stop, pause, rewind, fast-forward, show, and hide a video clip.

To control a video clip with a behavior, use the Behaviors panel to apply the behavior to a triggering object, such as a movie clip. Specify the event that triggers the behavior (such as releasing the movie clip), select a target object (the video that is affected by the behavior), and when necessary, select settings for the behavior, such as the number of frames to rewind.

Note: The triggering object must be a movie clip. You cannot attach video playback behaviors to button symbols or button components.

The following behaviors in Flash control embedded video:

Behavior	Purpose	Parameters
Play Video	Plays a video in the current document.	Instance name of target video
Stop Video	Stops the video.	Instance name of target video
Pause Video	Pauses the video.	Instance name of target video
Rewind Video	Rewinds the video by the specified number of frames.	Instance name of target video Number of frames
Fast Forward Video	Fast-forwards the video by the specified number of frames.	Instance name of target video Number of frames
Hide Video	Hides the video.	Instance name of target video
Show Video	Shows the video.	Instance name of target video

Control video playback using behaviors

- 1 Select the movie clip to trigger the behavior.
- 2 In the Behaviors panel (Window > Behaviors), click the Add (+) button, and select the desired behavior from the Embedded Video submenu.
- 3 Select the video to control.
- 4 Select a Relative or Absolute path.
- 5 If required, select settings for the behavior parameters and click OK.
- 6 In the Behaviors panel under Event, click On Release (the default event) and select a mouse event. To use the On Release event, leave the option unchanged.

See also

[“Controlling instances with behaviors”](#) on page 169

[“Control video playback using the Timeline”](#) on page 292

The FLVPlayback component

The FLVPlayback component lets you include a video player in your Flash application to play progressively downloaded video (FLV or F4V) files over HTTP, or play streaming FLV files from Flash Media Server (FMS) or a Flash Video Streaming Service (FVSS).

The FLVPlayback component does the following:

- Provides a set of prefabricated skins to customize playback controls and the look and feel of the user interface.
- Lets advanced users create their own custom skins.
- Provides cue points to synchronize your video with the animation, text, and graphics in your Flash application.
- Provides live preview of customizations.
- Maintains a reasonably sized SWF file for easy download.

The FLVPlayback component is the display area in which you view video. The FLVPlayback component includes the FLV Custom UI controls, a set of control buttons that play, stop, pause, and control playback the video.

Configure the FLVPlayback component

- 1 With the FLVPlayback component selected on the stage, open the Property inspector (Window > Properties) and enter an instance name.
- 2 Select Parameters in the Property inspector or open the Component inspector (Window > Components).
- 3 Enter values for parameters or use default settings.

For each FLVPlayback component instance you can set the following parameters in the Property inspector or in the Component inspector:

Note: *In most instances, it is not necessary to alter the settings in the FLVPlayback component unless you want to change the appearance of a video skin. The Video Import wizard sufficiently configures the parameters for most deployments.*

autoPlay Boolean value that determines how to play the FLV or F4V. If `true`, the video plays immediately when it is loaded. If `false`, loads the first frame and pauses. The default value is `true`.

autoRewind Boolean value that determines whether the video is automatically rewound. If `true`, the `FLVPlayback` component automatically rewinds the video to the beginning when the playhead reaches the end or when the user clicks the stop button. If `false`, the component does not automatically rewind the video. The default value is `true`.

autoSize Boolean value that, if `true`, resizes the component at runtime to use the source video dimensions. The default value is `false`.

Note: The encoded frame size of the video is not the same as the default dimensions of the FLVPlayback component.

bufferTime Number of seconds to buffer before beginning playback. The default value is 0.

contentPath (AS2 files) String that specifies the URL to an FLV, F4V, or to an XML file that describes how to play the video. Double-click the Value cell for this parameter to activate the Content Path dialog box. The default is an empty string. If you do not specify a value for the `contentPath` parameter, nothing happens when Flash executes the `FLVPlayback` instance.

source (AS3 files) String that specifies the URL to an FLV, F4V, or to an XML file that describes how to play the video. Double-click the Value cell for this parameter to activate the Content Path dialog box. The default is an empty string. If you do not specify a value for the `contentPath` parameter, nothing happens when Flash executes the `FLVPlayback` instance.

isLive Boolean value that, if `true`, specifies that the video is streaming live from FMS. The default value is `false`.

cuePoints A string that specifies the cue points for the video. Cue points allow you to synchronize specific points in the video with Flash animation, graphics, or text. The default value is an empty string.

maintainAspectRatio A Boolean value that, if `true`, resizes the video player in the `FLVPlayback` component to retain the source video aspect ratio; the source video is still scaled and the `FLVPlayback` component itself is not resized. The `autoSize` parameter takes precedence over this parameter. The default value is `true`.

skin A parameter that opens the Select Skin dialog box and allows you to choose a skin for the component. The default value is `None`. If you choose `None`, the `FLVPlayback` instance does not have control elements that allow the user to play, stop, or rewind the video, or take other actions that the controls make possible. If the `autoPlay` parameter is set to `true`, the video plays automatically. For more information, see “Customizing the `FLVPlayback` component” in *Using ActionScript 3.0 Components* or *ActionScript 2.0 Components Language Reference*.

totalTime Total number of seconds in the source video. The default value is 0. If you use progressive download, Flash uses this number if it is set to a value greater than zero (0). Otherwise, Flash tries to take the time from metadata.

Note: If you're using FMS or FVSS, this value is ignored; the total time of the video is taken from the server.

volume A number from 0 to 100 that represents the percentage of maximum volume at which to set the volume.

Specify the contentPath or source parameter

If you imported a local video clip into Flash for use with progressively downloaded or streaming video content, update the `contentPath` (AS2 FLA files) or `source` (AS3 FLA files) parameter of the `FLVPlayback` component before uploading your content to a web server or Flash Media Server. The `contentPath` or `source` parameter specifies the name and location of the video file on the server, and implies the playback method (for example, progressively downloading using HTTP, or streaming from Flash Media Server using RTMP).

- 1 With the `FLVPlayback` component selected on the Stage, open the Property inspector (Window > Properties) and select Parameters in the Property inspector, or open the Component inspector (Window > Component Inspector).
- 2 Enter values for parameters, or use the default settings as appropriate. For the `contentPath` or `source` parameter, do the following:
 - a Double-click the Value cell for the `contentPath` or `source` parameter to activate the Content Path dialog box.

- b Enter the URL or local path to the FLV or F4V file, or the XML file (for Flash Media Server or FVSS) that describes how to play the video.

If you do not know the location of the video or XML file, click the folder icon to navigate to the correct location. When browsing for an video file, if it is at or below the location of the target SWF file, Flash automatically makes the path relative to that location so that it is ready for serving from a web server. Otherwise, it is an absolute Windows or Macintosh file path.

If you specify an HTTP URL, the video file is a progressive download FLV or F4V file. If you specify a URL that is a Real-Time Messaging Protocol (RTMP) URL, the video streams from a Flash Media Server (FMS). A URL to an XML file could also be a streaming video file from FMS or from a FVSS.

Note: When you click OK on the Content Path dialog box, Flash updates the value of the `cuePoints` parameter, too, because you might have changed the `contentPath` parameter so that the `cuePoints` parameter no longer applies to the current content path. As a result, you lose any disabled cue points, although not ActionScript cue points. For this reason, you may want to disable non-ActionScript cue points through ActionScript, rather than through the Cue Points dialog box.

When you specify the `contentPath` or `source` parameter, Flash attempts to verify that the video you specified is compatible with Flash Player. If you see a warning dialog box, try re-encoding the video to FLV or F4V format with Adobe Media Encoder.

You can also specify the location of an XML file that describes how to play multiple video streams for multiple bandwidths. The XML file uses Synchronized Multimedia Integration Language (SMIL) to describe the video files. For a description of the XML SMIL file, see “Using a SMIL file” in the *ActionScript 2.0 Components Language Reference*.

Media components (Flash Player 6 and 7)

Note: The media components were introduced in Macromedia Flash MX Professional 2004 and are intended for use with Flash Player 6 or 7. If you are developing video content to use with Flash Player 8, instead use the `FLVPlayback` component introduced in Macromedia Flash Professional 8. The `FLVPlayback` component provides improved functionality, giving you more control over video playback in the Flash environment.

The media component suite consists of three components: `MediaDisplay`, `MediaController`, and `MediaPlayback`. With the `MediaDisplay` component, to add media to your Flash documents, drag the component to the Stage and configure it in the Component inspector. In addition to setting the parameters in the Component inspector, you can add cue points to trigger other actions. The `MediaDisplay` component has no visual representation during playback; only the video clip is visible.

The `MediaController` component provides user interface controls that let the user interact with streaming media. The Controller features Play, Pause, and Rewind to Start buttons and a volume control. It also includes playbars that show how much of the media has loaded and how much has played. A playhead slider can be dragged forward and backward on the playbar to navigate quickly to different parts of the video. Using behaviors or ActionScript, you can easily link this component to the `MediaDisplay` component to show streaming video and provide user control.

The `MediaPlayback` component provides the easiest and quickest way to add video and a controller to your Flash documents. The `MediaPlayback` component combines the `MediaDisplay` and `MediaController` components into a single, integrated component. The `MediaDisplay` and `MediaController` component instances are automatically linked to each other for playback control.

To configure parameters for playback, size, and layout for all three components, use the Component inspector or the Parameters tab in the Property inspector. All the media components work equally well with mp3 audio content.

For more information on the media components, “Media components,” in the *ActionScript 2.0 Components Language Reference*.

Chapter 12: Creating e-learning content

With Adobe® Flash® CS4 Professional, you can create e-learning content such as online courses and instructional materials. You can develop e-learning content manually or use pre-made components and learning interactions from the Flash community.

Getting started with Flash e-learning

About Flash e-learning

With the Flash authoring environment, you can create interactive online (e-learning) courses that run in Flash Player. You can manually create this content on your own, or use pre-made components and learning interactions available from third-party companies and the Flash community. Learning interactions include true/false responses, multiple choice questions, fill-in-the-blank answers, etc.

The following are some of the benefits of creating e-learning content in Flash:

- Anyone with a Flash-enabled web browser or the Adobe® AIR™ runtime can use the instructional content.
- You can customize the e-learning application interface to meet your needs and create high-quality interfaces that load quickly and have a consistent look on different platforms.
- Third-party learning interactions added to your online course provide a simple interface for entering question response data without writing ActionScript® code.
- Third-party Flash learning interactions can send tracking information to a server-side learning management system (LMS) that complies with the Aviation Industry CBT Committee (AICC) protocol or Shareable Content Object Reference Model (SCORM) standards.

System requirements

E-learning courseware runs on any computer with Flash® Player 6 or later and a Flash-enabled web browser or the Adobe® AIR™ runtime.

To track user data from the Flash learning interactions, you must have the following:

- A web server-side LMS, such as an AICC- or SCORM-compatible system
- A Flash Player-compatible web browser or the Adobe® AIR™ runtime. For a list of Flash Player compatible browsers, see www.adobe.com/go/flash_player_sysreqs_en.

Note: Tracking to an LMS with learning interactions does not work with Internet Explorer on the Macintosh.

Chapter 13: Creating accessible content

With the accessibility features in Adobe® Flash® CS4 Professional, you can create content that users with disabilities can use and enjoy.

About accessible content

Accessibility overview

You can create content that is accessible to all users, including those with disabilities, using the accessibility features that Adobe® Flash® CS4 Professional provides in the authoring environment user interface, taking advantage of ActionScript® designed to implement accessibility. As you design accessible Flash applications, consider how users might interact with the content and follow recommended design and development practices.

For a tutorial about accessible content, see Create Accessible Flash Content on the Flash Tutorials page at www.adobe.com/go/learn_fl_tutorials.

For a sample of accessible rich media content, see the Flash Samples page at www.adobe.com/go/learn_fl_samples. Download and decompress the Samples zip file and navigate to the Accessibility\AccessibleApplications folder to access the sample.

For the latest information on creating and viewing accessible Flash content, including supported platforms, screen reader compatibility, articles, and accessible examples, see the Flash Accessibility web page at www.adobe.com/go/flash_accessibility/.

Worldwide accessibility standards

Many countries have adopted accessibility standards based on the standards developed by the World Wide Web Consortium (W3C). The W3C publishes the *Web Content Accessibility Guidelines*, a document that prioritizes actions designers should take to make web content accessible. For information about the Web Accessibility Initiative, see the W3C website at w3.org.

In the United States, the law that governs accessibility is commonly known as Section 508, which is an amendment to the U.S. Rehabilitation Act.

For additional information about Section 508, see the following websites:

- The US government-sponsored website at section508.gov
- The Adobe accessibility site at www.adobe.com/accessibility/

Understanding screen reader technology

Screen readers are software applications that visually impaired users can use to navigate a website and read the web content aloud. To enable a screen reader to read nontextual objects in your application, such as vector art and animations, use the Accessibility panel to associate a name and description with the object. The keyboard shortcuts you define can allow users to use the screen reader to navigate through your document with ease.

To expose graphic objects, use the Accessibility panel or ActionScript to provide a description.

You cannot control how any screen reader behaves; you can control only the content, which you can mark up in your Flash applications to expose the text and ensure that screen reader users can activate the controls. You decide which objects in the Flash application are exposed to screen readers, provide descriptions for them, and decide the order in which they are exposed to screen readers. You cannot force screen readers to read specific text at specific times or control the manner in which that content is read. Test your applications with a variety of screen readers to ensure that they perform as you expect.

Sound is the most important medium for most screen reader users. Consider how any sound in your document interacts with the text spoken aloud by screen readers. It might be difficult for screen reader users to hear what their screen readers are saying if your Flash application contains loud sounds.

Platform requirements

You can only create Flash content designed for use with screen readers with Windows platforms. Viewers of Flash content must have Macromedia Flash® Player 6 from Adobe or later and Internet Explorer on Windows 98 or later.

See also

“Create a keyboard shortcut to an object for screen readers” on page 311

“Using Flash to enter accessibility information for screen readers” on page 305

Flash and Microsoft Active Accessibility (Windows only)

Flash Player is optimized for Microsoft Active Accessibility (MSAA), which provides a descriptive and standardized way for applications and screen readers to communicate. MSAA is available only for Windows operating systems. For more information on Microsoft Accessibility Technology, visit the Microsoft Accessibility website at www.microsoft.com/enable/default.aspx.

The Windows ActiveX (Internet Explorer plug-in) version of Flash Player 6 supports MSAA, but Windows Netscape and Windows stand-alone players do not.

Important: *MSAA is currently not supported in the opaque windowless and transparent windowless modes. (These modes are options in the HTML Publish Settings panel, available for use with the Windows version of Internet Explorer 4.0 or later, with the Flash ActiveX control.) To make your Flash content accessible to screen readers, avoid using these modes.*

Flash Player makes information about the following types of accessibility objects available to screen readers that use MSAA.

Dynamic or static text The principal property of a text object is its name. To comply with MSAA conventions, the name is equal to the contents of the text string. A text object can also have an associated description string. Flash uses the static or dynamic text immediately above or to the left of an input text field as a label for that field.

Note: *Any text that is a label is not passed to a screen reader, but is used as the name of the object that it labels. Labels are never assigned to buttons or text fields that have author-supplied names.*

Input text fields Have a value, an optional name, a description string, and a keyboard shortcut string. An input text object’s name can come from a text object that is above or to the left of it.

Buttons Have a state (pressed or not pressed), support a programmatic default action that causes the button to depress momentarily, and optionally have a name, a description string, and a keyboard-shortcut string. Flash uses any text entirely inside a button as a label for that button.

Note: *For accessibility purposes, Flash Player considers movie clips used as buttons with button event handlers such as `onPress` to be buttons, not movie clips.*

Components Provide special accessibility implementation.

Movie clips Exposed to screen readers as graphic objects when they do not contain any other accessible objects, or when you use the Accessibility panel to provide a name or a description for a movie clip. When a movie clip contains other accessible objects, the clip itself is ignored, and the objects inside it are made available to screen readers.

Note: All Flash Video objects are treated as simple movie clips.

See also

“Using Flash to enter accessibility information for screen readers” on page 305

“Using accessible components” on page 314

“Creating accessibility with ActionScript” on page 312

Basic accessibility support in Flash Player

By default, the following objects are defined as accessible in all Flash documents and are included in the information that Flash Player provides to screen reader software. This generic support for documents that do not use any accessibility features includes the following:

Dynamic or static text Text is transferred to the screen reader program as a name, but with no description.

Input text fields Text is transferred to the screen reader. No names are transferred, except where a labeling relationship is found for the input text, such as a static text field positioned close to the input text field. No descriptions or keyboard shortcut strings are transferred.

Buttons The state of the button is transferred to the screen reader. No names are transferred, except where labeling relationships are found, and no descriptions or keyboard shortcut strings are transferred.

Documents The document state is transferred to the screen reader, but with no name or description.

Accessibility for hearing-impaired users

Include captions for audio content that is integral to comprehending the material. A video of a speech, for example, might require captions for accessibility, but a quick sound associated with a button probably wouldn't.

Methods to add captions to a Flash document include the following:

- Add text as captions, ensuring that the captions are synchronized with the audio in the Timeline.
- Use Hi-Caption Viewer, a component available from Hi Software that works with Hi-Caption SE for use with Flash (see www.adobe.com/go/accessible_captions). *Captioning Macromedia Flash Movies with Hi-Caption SE*, a white paper, explains how to use Hi-Caption SE and Flash together to create a captioned document (see www.adobe.com/go/accessibility_papers).

Provide animation accessibility for the visually impaired

You can change the property of an accessible object during SWF file playback. For example, to indicate changes that take place on a keyframe in an animation. However, different vendor's screen readers treat new objects on frames differently. Some screen readers might read only the new object, whereas other screen readers might re-read the entire document.

To reduce the chance of causing a screen reader to emit extra “chatter” that can annoy users, avoid animating the text, buttons, and input text fields in your document. Also, avoid making your content loop.

Flash Player can't determine the actual text content of features such as Text Break Apart to animate text. Screen readers can only provide accurate accessibility to information-carrying graphics such as icons and gestural animation, if you provide names and descriptions for these objects in your document or for the entire Flash application. You can also add supplementary text to your document or shift important content from graphics to text.

- 1 Select the object for which you want to change the accessibility properties.
- 2 Select Window > Other Panels > Accessibility.
- 3 Change the properties for the object.

Alternatively, use ActionScript to update accessibility properties.

See also

[“Make an entire SWF application accessible”](#) on page 308

[“Creating accessibility with ActionScript”](#) on page 312

Testing accessible content

When you test your accessible Flash applications, follow these recommendations:

- Download several screen readers and test your application by playing it in a browser with the screen reader enabled. Check that the screen reader is not attempting to “talk over” places in your document where you inserted separate audio. Several screen reader applications provide a demonstration version of the software as a free download; test as many screen readers as you can to ensure compatibility across screen readers.
- Test interactive content and verify that users can navigate your content effectively using only the keyboard. Different screen readers work in different ways when processing input from the keyboard; your Flash content might not receive keystrokes as you intended. Test all keyboard shortcuts.

Using Flash to enter accessibility information for screen readers

Flash for screen readers and accessibility

Screen readers read aloud a description of the content, read text, and assist users as they navigate through the user interfaces of traditional applications such as menus, toolbars, dialog boxes, and input text fields.

By default, the following objects are defined as accessible in all Flash documents and are included in the information that Flash Player provides to screen reader software:

- Dynamic text
- Input text fields
- Buttons
- Movie clips
- Entire Flash applications

Flash Player automatically provides names for static and dynamic text objects, which are the contents of the text. For each of these accessible objects, you can set descriptive properties for screen readers to read aloud. You can also control how Flash Player decides which objects to expose to screen readers—for example, you can specify that certain accessible objects are not exposed to screen readers at all.

The Flash Accessibility panel

The Flash Accessibility panel (Window > Other Panels > Accessibility) lets you provide accessibility information to screen readers and set accessibility options for individual Flash objects or entire Flash applications.

Note: Alternatively, use ActionScript code to enter accessibility information.

If you select an object on the Stage, you can make that object accessible and specify options and tab order for the object. For movie clips, you can specify whether child object information is passed to the screen reader (the default when you make an object accessible).

With no objects selected on the Stage, use the Accessibility panel to assign accessibility options for an entire Flash application. You can make the entire application accessible, make child objects accessible, have Flash label objects automatically, and give specific names and descriptions to objects.

All objects in Flash documents must have instance names for you to apply accessibility options to them. Create instance names for objects in the Property inspector. The instance name is used to refer to the object in ActionScript.

The following options are available in the Accessibility panel:

Make Object Accessible (Default) Instructs Flash Player to pass the accessibility information for an object to a screen reader. When the option is disabled, accessibility information for the object is not passed to screen readers. Disabling this option as you test content for accessibility can be useful because some objects might be extraneous or decorative and making them accessible could produce confusing results in the Screen Reader. You can then apply a name manually to the labeled object, and hide the labeling text by deselecting Make Object Accessible. When Make Object Accessible is disabled, all other controls on the Accessibility panel are disabled.

Make Child Objects Accessible (Movie clips only; Default) Instructs Flash Player to pass child object information to the screen reader. Disabling this option for a movie clip causes that movie clip to appear as a simple clip in the accessible object tree, even if the clip contains text, buttons, and other objects. All objects in the movie clip are then hidden from the object tree. This option is useful mainly for hiding extraneous objects from screen readers.

Note: If a movie clip is used as a button—it has a button event handler assigned to it, such as `onPress` or `onRelease`—the Make Child Objects Accessible option is ignored because buttons are always treated as simple clips, and their children are never examined, except in the case of labels.

Auto Label Instructs Flash to automatically label objects on the Stage with the text associated with them.

Name Specifies the object name. Screen readers identify objects by reading these names aloud. When accessible objects don't have specified names, a screen reader might read a generic word, such as *Button*, which can be confusing.

Important: Do not confuse object names specified in the Accessibility panel with instance names specified in the Property inspector. Giving an object a name in the Accessibility panel does not give it an instance name.

Description Lets you enter a description of the object to the screen reader. The screen reader reads this description.

Shortcut Describes keyboard shortcuts to the user. The screen reader reads the text in this text field. Entering keyboard shortcut text here does not create a keyboard shortcut for the selected object. You must provide ActionScript keyboard handlers to create shortcut keys.

Tab Index (Adobe® Flash® CS4 Professional only) Creates a tab order in which objects are accessed when the user presses the tab key. The tab index feature works for keyboard navigation through a page, but not for screen reader reading order.

For more information, see the Flash Accessibility web page at www.adobe.com/go/flash_accessibility/.

For a tutorial about accessible content, see Create Accessible Flash Content on the Flash Tutorials page at www.adobe.com/go/learn_fl_tutorials.

For a sample of accessible rich media content, see the Flash Samples page at www.adobe.com/go/learn_fl_samples. Download and decompress the Samples zip file and navigate to the Accessibility\AccessibleApplications folder to access the sample.

See also

“[Creating accessibility with ActionScript](#)” on page 312

“[Create a keyboard shortcut to an object for screen readers](#)” on page 311

“[Turn off automatic labeling and specify an object name for screen readers](#)” on page 310

Selecting names for buttons, text fields, and entire SWF applications

Use the Accessibility panel in the following ways to assign names to buttons and input text fields so that the screen reader identifies them appropriately:

- Use the auto label feature to assign text adjacent or in the object as a label.
- Enter a specific label in the Accessibility panel name field.

Flash automatically applies the name that you place on top of, in, or near a button or text field as a text label. Labels for buttons must appear within the bounding shape of the button. For the button in the following example, most screen readers would first read the word *button*, then read the text label *Home*. The user can press Return or Enter to activate the button.



A form might include an input text field where users enter their names. A static text field, with the text *Name* appears next to the input text field. When Flash Player discovers such an arrangement, it assumes that the static text object serves as a label for the input text field.

For example, when the following part of a form is encountered, a screen reader reads “Enter your name here.”



A. Static text B. Input text field

In the Accessibility panel, turn off automatic labeling if it is not appropriate for your document. You can also turn off automatic labeling for specific objects in your document.

See also

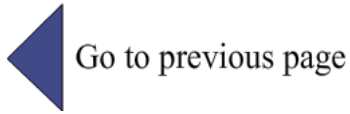
“[Turn off automatic labeling and specify an object name for screen readers](#)” on page 310

Provide a name for an object

You can turn off automatic labeling for part of an application and provide names for the objects in the Accessibility panel. If you have automatic labeling turned on, you can select specific objects and provide names for the objects in the Name text field in the Accessibility panel so that the name is used instead of the object text label.

When a button or input text field doesn’t have a text label, or when the label is in a location that Flash Player can’t detect, you can specify a name for the button or text field. You can also specify a name if the text label is near a button or text field, but you don’t want that text to be used as that object’s name.

In the following example, the text that describes the button appears outside and to the right of the button. In this location, Flash Player does not detect the text, and the screen reader does not read it.



To correct this situation, open the Accessibility panel, select the button, and enter the name and description. To prevent repetition, make the text object inaccessible.

***Note:** An object's accessibility name is unrelated to the ActionScript instance name or ActionScript variable name associated with the object. (This information generally applies to all objects.) For information on how ActionScript handles instance names and variable names in text fields, see *About text field instance and variable names in Learning ActionScript 2.0 in Adobe Flash* at www.adobe.com/go/learn_fl_cs4_learningAS2_en.*

Specify a name and description for a button, text field, or entire SWF application

- 1 Do one of the following:
 - To provide a name for a button or text field, select the object on the Stage.
 - To provide a name for an entire Flash application, deselect all objects on the Stage.
- 2 Select Window > Other Panels > Accessibility.
- 3 Select either Make Object Accessible (for buttons or text fields) or the default, Make Movie Accessible (for entire Flash applications).
- 4 Enter a name and description for the button, text field, or Flash application.

Define accessibility for a selected object in a SWF application

- 1 Select Window > Other Panels > Accessibility.
- 2 Do one of the following:
 - Select Make Object Accessible (the default setting) to expose the object to screen readers and to enable other options in the panel.
 - Deselect Make Object Accessible to hide the object from screen readers and disable the other options in the panel.
- 3 Enter a name and a description for the selected object as needed:

Dynamic text To provide a description for static text, you must convert it to dynamic text.

Input text fields or buttons Enter a keyboard shortcut.

Movie clips Select Make Child Objects Accessible to expose the objects inside the movie clip to screen readers.

***Note:** If you can describe your application in a simple phrase that a screen reader can easily convey, turn off Make Children Accessible, and type a suitable description.*

Make an entire SWF application accessible

After a Flash document is complete and ready to be published or exported, make the entire Flash application accessible.

- 1 Deselect all elements in the document.
- 2 Select Window > Other Panels > Accessibility.

- 3 Select Make Movie Accessible (the default setting) to expose the document to screen readers.
- 4 Select or deselect Make Children Accessible to expose or omit any accessible objects in the document to screen readers.
- 5 If you selected Make Movie Accessible in step 2, enter a name and description for the document as needed.
- 6 Select Auto Label (the default setting) to use text objects as automatic labels for accessible buttons or input text fields contained in the document. Deselect this option to turn off automatic labeling and expose text objects to screen readers as text objects.

Viewing and creating tab order and reading order

The two aspects to tab indexing order are the *tab order* in which a user navigates through the web content and the order in which things are read by the screen reader, called the *reading order*.

Flash Player uses a tab index order from left to right and top to bottom. Customize both the tab and reading order by using the `tabIndex` property in ActionScript (in ActionScript, the `tabIndex` property is synonymous with the reading order).

Note: *Flash Player no longer requires that you add all of the objects in a FLA file to a list of tab index values. Even if you do not specify a tab index for all objects, a screen reader reads each object correctly.*

Tab order The order in which objects receive input focus when users press the Tab key. Use ActionScript to create the tab order, or if you have Adobe® Flash® CS4 Professional, use the Accessibility panel. The tab index that you assign in the Accessibility panel does not necessarily control the reading order.

Reading order The order in which a screen reader reads information about the object. To create a reading order, use ActionScript to assign a tab index to every instance. Create a tab-order index for every accessible object, not just the focusable objects. For example, dynamic text must have tab indexes, even though a user cannot tab to dynamic text. If you do not create a tab index for every accessible object in a given frame, Flash Player ignores all tab indexes for that frame whenever a screen reader is present, and uses the default tab ordering instead.

See also

[“Use ActionScript to create a tab order for accessible objects”](#) on page 314

Create a tab-order index for keyboard navigation in the Accessibility panel

You can create a custom tab-order index in the Accessibility panel for keyboard navigation for the following objects:

- Dynamic text
- Input text
- Buttons
- Movie clips, including compiled movie clips
- Components
- Screens

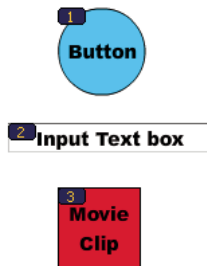
Note: *You can also use ActionScript code to create a tab-order index for keyboard navigation.*

Tab focus occurs in numerical order, starting from the lowest index number. After tab focus reaches the highest tab index, focus returns to the lowest index number.

When you move tab-indexed objects that are user-defined in your document, or to another document, Flash retains the index attributes. Check for and resolve index conflicts (for example, two different objects on the Stage with the same tab-index number).

Important: *If two or more objects have the same tab index in any given frame, Flash follows the order in which the objects were placed on the Stage.*

- 1 Select the object in which to assign a tab order.
- 2 Select Window > Other Panels > Accessibility.
- 3 If you're providing an index for the selected object only, in the Tab Index text field, enter a positive integer (up to 65535) that reflects the order in which the selected object should receive focus.
- 4 To view a tab order, select View > Show Tab Order. Tab index numbers for individual objects appear in the upper-left corner of the object.



Tab index numbers

Note: *Tab indexes created with ActionScript code do not appear on the Stage when the Show Tab Order option is enabled.*

Specifying advanced accessibility options for screen readers

Turn off automatic labeling and specify an object name for screen readers

- 1 On the Stage, select the button or input text field for which you want to control labeling.
- 2 Select Window > Other Panels > Accessibility.
- 3 Select Make Object Accessible (the default setting).
- 4 Enter a name for the object. The name is read as the label for the button or text field.
- 5 To turn off accessibility for the automatic label (and hide it from screen readers), select the text object on the Stage.
- 6 If the text object is static text, convert it to dynamic text (in the Property inspector, select Text type > Dynamic Text).
- 7 Deselect Make Object Accessible.

Hide an object from the screen reader

You can hide a selected object from screen readers, and you can decide to hide accessible objects that are contained inside a movie clip or Flash application and expose only the movie clip or Flash application to screen readers.

Note: Only hide objects that are repetitive or convey no content.

When an object is hidden, the screen reader ignores the object.

- 1 On the Stage, select the button or input text field to hide from the screen reader.
- 2 Select Window > Other Panels > Accessibility.
- 3 In the Accessibility panel, do one of the following:
 - If the object is a movie clip, button, text field, or another object, deselect Make Object Accessible.
 - If the object is the child of a movie clip, deselect Make Child Objects Accessible.

Create a keyboard shortcut to an object for screen readers

You can create a keyboard shortcut for an object, such as a button, so users can navigate to it without listening to the contents of an entire page. For example, you can create a keyboard shortcut to a menu, a toolbar, the next page, or a submit button.

To create a keyboard shortcut, write ActionScript code for an object. If you provide a keyboard shortcut for an input text field or button, you must also use the ActionScript Key class to detect the key the user presses during Flash content playback. See Key in the *ActionScript 2.0 Language Reference*. See Capturing keypresses in *Learning ActionScript 2.0 in Adobe Flash* at www.adobe.com/go/learn_fl_cs4_learningAS2_en.

Select the object and add the name of the keyboard shortcut to the Accessibility panel so the screen reader can read it.

Test your Flash content with multiple screen readers. Keyboard shortcut functionality also depends on the screen reader software used. The key combination Control+F, for example, is a reserved keystroke for both the browser and the screen reader. The screen reader reserves the arrow keys. Generally, you can use the 0 to 9 keys on the keyboard for keyboard shortcuts, however, screen readers increasingly use even these keys.

See also

“[Testing accessible content](#)” on page 305

Create a keyboard shortcut

- 1 On the Stage, select the button or input text field to create a keyboard shortcut for.
- 2 Select Window > Other Panels > Accessibility.
- 3 In the Shortcut field, type the name of the keyboard shortcut, using the following conventions:
 - Spell out key names, such as Control or Alt.
 - Use capital letters for alphabetic characters.
 - Use a plus sign (+) between key names, with no spaces (for example, Control+A).

Important: Flash does not check that the ActionScript to code the keyboard shortcut was created.

Map a keyboard shortcut to a button instance Control+7 to myButton instance

- 1 Select the object on the Stage, display the Accessibility panel, and in the Shortcut field, type the key combination of the shortcut. For example, `Control+7`.
- 2 Enter the following ActionScript 2.0 code in the Actions panel:

Note: In this example the shortcut is Control+7.

```
function myOnPress() {
    trace( "hello" );
}
function myOnKeyDown() {
    if (Key.isDown(Key.CONTROL) && Key.getCode() == 55) // 55 is key code for 7
    {
        Selection.setFocus(myButton);
        myButton.onPress();
    }
}
var myListener = new Object();
myListener.onKeyDown = myOnKeyDown;
Key.addListener(myListener);
myButton.onPress = myOnPress;
myButton._accProps.shortcut = "Ctrl+7"
Accessibility.updateProperties();
```

Note: The example assigns the Control+7 keyboard shortcut to a button with an instance name of myButton and makes information about the shortcut available to screen readers. In this example, when you press Control+7, the myOnPress function displays the text “hello” in the Output panel. See *addListener (IME.addListener method)* in *ActionScript 2.0 Language Reference* at www.adobe.com/go/learn_fl_cs4_as2lr_en.

Creating accessibility with ActionScript

About ActionScript and accessibility

You can create accessible documents with ActionScript® code. For accessibility properties that apply to the entire document, you can create or modify a global variable called `_accProps`. See the `_accProps` property in *ActionScript 2.0 Language Reference* at www.adobe.com/go/learn_fl_cs4_as2lr_en.

For properties that apply to a specific object, you can use the syntax `instancename._accProps`. The value of `_accProps` is an object that can include any of the following properties:

Property	Type	Equivalent selection in the Accessibility panel	Applies to
<code>.silent</code>	Boolean	Make Movie Accessible/Make Object Accessible (inverse logic)	Entire documents Buttons Movie clips Dynamic text Input text
<code>.forceSimple</code>	Boolean	Make Child Objects Accessible (inverse logic)	Entire documents Movie clips

Property	Type	Equivalent selection in the Accessibility panel	Applies to
.name	string	Name	Entire documents Buttons Movie clips Input text
.description	string	Description	Entire documents Buttons Movie clips Dynamic text Input text
.shortcut	string	Shortcut	Buttons Movie clips Input text

Note: With inverse logic, a value of `true` in ActionScript corresponds to a check box that is not selected in the Accessibility panel, and a value of `false` in ActionScript corresponds to a selected check box in the Accessibility panel.

Modifying the `_accProps` variable has no effect by itself. You must also use the `Accessibility.updateProperties` method to inform screen reader users of Flash content changes. Calling the method causes Flash Player to re-examine all accessibility properties, update property descriptions for the screen reader, and, if necessary, send events to the screen reader that indicate changes have occurred.

When updating accessibility properties of multiple objects at once, include only a single call to `Accessibility.updateProperties` (too frequent updates to the screen reader can cause some screen readers to become too verbose).

See the `Accessibility.updateProperties` method in the *ActionScript 2.0 Language Reference* at www.adobe.com/go/learn_fl_cs4_as2lr_en.

Implementing screen reader detection with the `Accessibility.isActive()` method

To create Flash content that behaves in a specific way if a screen reader is active, use the `Accessibility.isActive()` ActionScript method, which returns a value of `true` if a screen reader is present, and `false` otherwise. You can then design your Flash content to perform so that it's compatible with screen reader use (for example, by hiding child elements from the screen reader). For more information, see the `Accessibility.isActive` method in the *ActionScript 2.0 Language Reference* at www.adobe.com/go/learn_fl_cs4_as2lr_en.

For example, you could use the `Accessibility.isActive()` method to decide whether to include unsolicited animation. Unsolicited animation happens without the screen reader doing anything, which can be confusing for screen readers.

The `Accessibility.isActive()` method provides asynchronous communication between the Flash content and Flash Player; a slight real-time delay can occur between the time the method is called and the time when Flash Player becomes active, returning an incorrect value of `false`. To ensure that the method is called correctly, do one of the following:

- Instead of using the `Accessibility.isActive()` method when the Flash content first plays, call the method whenever you need to make a decision about accessibility.

- Introduce a short delay of one or two seconds at the beginning of your document to give the Flash content enough time to contact Flash Player.

For example, you can use an `onFocus` event to attach this method to a button. This approach generally gives the SWF file enough time to load and you can assume a screen reader user will tab to the first button or object on the Stage.

Use ActionScript to create a tab order for accessible objects

To create the tab order with ActionScript® code, assign the `tabIndex` property to the following objects:

- Dynamic text
- Input text
- Buttons
- Movie clips, including compiled movie clips
- Timeline frames
- Screens

Provide a complete tab order for all accessible objects. If you create a tab order for a frame and you don't specify a tab order for an accessible object in the frame, Flash Player ignores all the custom tab-order assignments. Additionally, all objects assigned to a tab order, except frames, must have an instance name specified in the Instance Name text field of the Property inspector. Even items that are not tab stops, such as text, need to be included in the tab order if they are to be read in that order.

Because static text cannot be assigned an instance name, it cannot be included in the list of the `tabIndex` property values. As a result, a single instance of static text anywhere in the SWF file causes the reading order to revert to the default.

To specify a tab order, assign an order number to the `tabIndex` property, as the following example shows:

```
_this.myOption1.btn.tabIndex = 1  
_this.myOption2.txt.tabIndex = 2
```

See `tabIndex` in `Button`, `MovieClip`, and `TextField` in the *ActionScript 2.0 Language Reference* at www.adobe.com/go/learn_fl_cs4_as2lr_en.

You can also use the `tabChildren()` or `tabEnabled()` methods to assign custom tab order. See `MovieClip.tabChildren`, `MovieClip.tabEnabled`, and `TextField.tabEnabled` in the *ActionScript 2.0 Language Reference* at www.adobe.com/go/learn_fl_cs4_as2lr_en.

See also

“Create a tab-order index for keyboard navigation in the Accessibility panel” on page 309

Using accessible components

A core set of UI components accelerates building accessible applications. These components automate many of the most common accessibility practices related to labeling, keyboard access, and testing and help ensure a consistent user experience across rich applications. Flash includes the following set of accessible components:

- `SimpleButton`
- `CheckBox`
- `RadioButton`

- Label
- TextInput
- TextArea
- ComboBox
- ListBox
- Window
- Alert
- DataGrid

Accessible Flash components must contain ActionScript that defines their accessible behavior. For information on which accessible components work with screen readers, see the Flash Accessibility web page at www.adobe.com/go/flash_accessibility/.

For general information about components, see “About Components” in *Using ActionScript 2.0 Components* at www.adobe.com/go/go/learn_fl_cs4_as2components_en.

For each accessible component, enable the accessible portion of the component with the `enableAccessibility()` command. This command includes the accessibility object with the component as the document is compiled. Because no simple way exists to remove an object after it is added to the component, these options are disabled by default. Therefore, it’s important that you enable accessibility for each component. Perform this step only once for each component; you do not need to enable accessibility for each instance of a component for a given document. See “Button component”, “CheckBox component”, “ComboBox component”, “Label component”, “List component”, “RadioButton component”, and “Window component” in the *ActionScript 2.0 Components Language Reference* at www.adobe.com/go/learn_fl_cs4_as2lr_en.

Chapter 14: Working with screens

Screens make it easy to create presentations, slide shows, and other slide-based content in Adobe® Flash® CS4 Professional. The Screens feature in Flash does not support ActionScript® 3.0. To use screens, you must start with an ActionScript 2.0-based FLA file.

Screen-based documents and the screen authoring environment

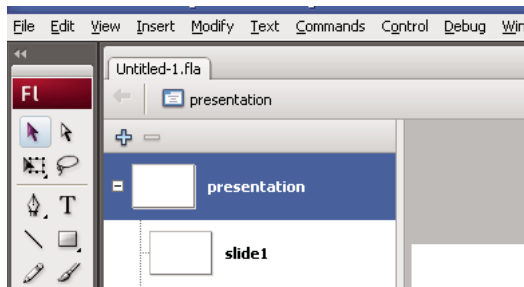
About the screen authoring environment

Screens provide an authoring user interface with structural building blocks that make it easy for you to create complex, hierarchical Flash documents, such as slide presentations or form-based applications.

Note: The Screens feature in Flash does not support ActionScript 3.0. To use screens, you must start with an ActionScript 2.0-based FLA file.

Screens let you structure complex applications without using multiple frames and layers in the Timeline, and without viewing the Timeline.

When you author a screen-based document, the screens are arranged in a structured hierarchy that you create. To structure the document, nest screens in a branching tree. You can easily preview and modify the structure of a screen-based document.



Detail of default pasteboard for a new Flash Slide Presentation. Screen thumbnails appear in the Screen Outline pane on the left side of the pasteboard, and the Timeline is collapsed.

To author a screen-based document, you first create a Slide Presentation or Form Application document. Then you add screens, configure the screens and add content, and add behaviors to create controls and transitions for the screens.

Slide presentations and screens, and form applications and screens

There are two types of screen-based documents. The type of document you select determines the type of default screen in the document.

- A Flash Slide Presentation uses the slide screen as the default screen type. A slide screen is designed for a sequential presentation.

- A Flash Form Application uses the form screen as the default screen type. A form screen is designed for a nonlinear, form-based application.

Although each document has a default screen type, you can include and mix slide screens and form screens in any screen-based document.

To control screens, set parameters for slide or form screens in the Property inspector or use ActionScript. For more information, see “Screen class”, “Form class”, and “Slide class”, in the [ActionScript 2.0 Components Language Reference](#).

Slide screens let you create Flash documents with sequential content, such as a slide show. The default runtime behavior lets users navigate sequentially through slide screens, using the left and right arrow keys. Sequential screens can overlay one another so that the previous screen remains visible when the next slide is viewed. Screens can continue playing after they are hidden. To automatically manage the visibility of each screen, use slide screens.

Form screens let you create structured form-based applications, such as online registration or e-commerce forms. By default, to create the navigation structure with form screens, you must write ActionScript code. To manage the visibility of individual screens yourself, use form screens.

To learn more about creating screen-based documents that use ActionScript, see “About organizing code for screens” in [Using ActionScript 2.0 Components](#).

See also

“[Creating controls and transitions for screens with behaviors](#)” on page 326

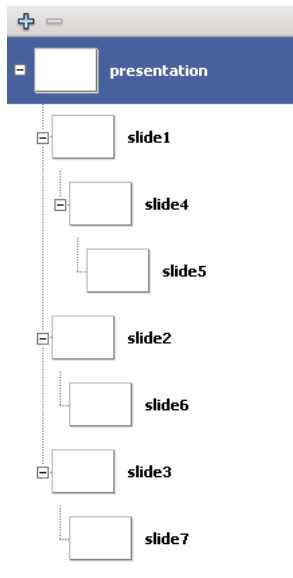
Document structure and hierarchy

Each document has a master screen at the top level. In a Flash Slide Presentation, the top-level screen is called Presentation by default. In a Flash Form Application, the top-level screen is called Application by default.

The top-level screen is the container for everything that you add to the document, including other screens. You can place content on the top-level screen. You cannot delete or move the top-level screen.

Screens are similar to nested movie clips in some ways: Child screens inherit the behavior of their parents, and you use target paths in ActionScript to send messages from one screen to another. However, screens do not appear in the library, and you cannot create multiple instances of a screen.

You can add multiple screens to a document, and you can nest screens within other screens, in as many levels as you want. A screen that is inside another screen is the *child* of that screen. A screen that contains another screen is the *parent* of that screen. If a screen is nested several layers deep, all the screens above that screen are its *ancestors*. Screens that are at the same level are *sibling* screens. All screens nested in another screen are its *descendants*. A child screen contains all the content of its ancestor screens.



Top-level slide has three children, Slide 1, Slide 2, and Slide 3. Slide 1 has one child and one grandchild

To learn more about creating screen-based documents that use ActionScript, see “About organizing code for screens in” in [Using ActionScript 2.0 Components](#).

Using preloaders with screen-based documents

To include a preloader (a separate SWF file that loads your primary SWF) with your screen-based document, create the preloader as a separate SWF file (non-screen-based), and load the SWF file for the screen-based document from within the preloader SWF.

You cannot create a preloader within a screen-based document, because all screens in a document are located on the first frame of the root Timeline, so you cannot call or load other frames.

Working with screens

Add screens to a document

You can add a default-type screen or select a screen type at the same level as the currently selected screen. The new screen is a *sibling* screen of the selected screen. You can also add a nested screen one level below the currently selected screen. To view all screens in a document, use the Screen Outline pane.

When you add screens to a document, Flash exhibits the following default behaviors:

- Uses the screen type of the document (slide type for a Slide Presentation or form type for a Form Application) for the new screen. To select to insert a screen of another type, use the Insert Screen Type command in the Screens context menu.
- Inserts the first screen you add directly after the top-level screen, one level below it.
- Inserts a screen after the currently selected screen, at the same level. If the document contains nested screens below the currently selected screen, the new screen is added after the nested screens, at the same level as the selected screen.

- Flash inserts a nested screen directly after the currently selected screen, and nested one level down. If the document contains a nested screen or screens below the currently selected screen, the new screen is added after all nested screens already in place, one level below the selected screen.

Add a default-type screen or a screen of a specified type at the current screen level

- 1 Select a screen in the Screen Outline pane.
- 2 To add a default-type screen, do one of the following:
 - Press Enter or Return.
 - Click the Insert Screen (+) button at the top of the Screen Outline pane.
 - Select Insert > Screen.
 - Select Insert Screen from the Screens context menu.
- 3 To add a screen of a specified type, select Insert Screen Type from the context menu and select a screen type.

Add a nested screen of the default type

- 1 Select a screen in the Screen Outline pane.
- 2 Do one of the following:
 - Press Enter or Return.
 - Select Insert > Nested Screen.
 - Select Insert Nested Screen from the Screens context menu.

Use the Screen Outline pane

The Screen Outline pane at the left of the Document window displays thumbnails of each screen in the current document, in a collapsible tree view. The tree represents the structural hierarchy of the document. Nested screens are indented below the screen that contains them.

When you add a screen to a document, the screen appears in the Screen Outline pane.

To hide and show nested screens, collapse and expand the tree. You can hide, show, and resize the Screen Outline pane.

- To display the screen on the Stage, click on a screen thumbnail in the Screen Outline pane.
- To hide or show the Screen Outline pane, select Window > Other Panels > Screens.
- To expand or collapse the tree, click the Plus (+) or Minus (-) button next to a screen to show or hide the screens nested within it (Windows), or click the triangle next to a screen to show or hide the screens nested within it (Macintosh).
- To resize the Screen Outline pane, drag the dividing line between the Screen Outline pane and the Document window.

See also

[“Creating controls and transitions for screens with behaviors”](#) on page 326

View the context menu for a screen

The Screens context menu contains commands for working with screens.

- ❖ Right-click (Windows) or Control-click (Macintosh) a screen thumbnail in the Screen Outline pane.

Using timelines with screens

Each screen in a screen-based Flash document has its own Timeline that is collapsed by default. To work with frames or layers, open the Timeline (Window > Timeline).

You cannot view or modify the root Timeline of a screen-based Flash document.

You can add frames, keyframes, and layers, and manipulate content on a screen's Timeline.

In the Timeline, nested screens work much as nested movie clips do, with some exceptions.

See also

[“How screens interact with ActionScript”](#) on page 328

Using the Movie Explorer with screens

To view and organize the contents of a document containing screens, use the Movie Explorer. The Movie Explorer handles documents that contain screens much as it handles documents that do not contain screens, with the following exceptions:

- The Movie Explorer shows only the contents of the current screen (the screen selected in the Screens Outline pane).
- You cannot view scenes in the Movie Explorer because a document with screens cannot contain scenes.

Select, move, and edit screens

When you select an individual screen in the Screen Outline pane, the screen appears in the Document window. To apply modifications to several screens at once, select multiple contiguous or discontinuous screens in the Screen Outline pane. When you select multiple screens, the contents of the first screen selected appear in the Screen Outline pane.

By default, the contents of a slide screen are not visible when you show the screen's parent in the Document window (the Hide Screen context menu option is selected). To show the contents of a slide screen when its parent appears, deselect this option. When the Hide Screen context menu option is deselected, you can select the child slide screen on the Stage. This feature affects display during authoring only, not runtime playback. The Hide Screen context menu option is deselected for form screens by default. To hide child form screens in the display during authoring, turn the option on.

You can cut, copy, paste, and drag screens in the Screen Outline pane to change their position in the document, and you can remove screens from a document.

*Note: The terms **child**, **parent**, and **ancestor** refer to the hierarchical relationships of nested screens.*

See also

[“Document structure and hierarchy”](#) on page 317

[“Speed up document display”](#) on page 437

View a screen in the Document window

❖ Do any of the following:

- To view that screen, click a screen thumbnail in the Screen Outline pane.
- To navigate to the screen, use the keyboard keys with the Screen Outline pane in focus.

- To navigate through the screens, select View > Go To and select the screen name from the submenu, or select First, Previous, Next, or Last.
- Click the Edit Screen button at the right side of the edit bar and select the screen name.

Select multiple screens in the Screen Outline pane

- To select multiple contiguous screens, Shift-click the first and the last screen to select.
- To select multiple discontinuous screens, Control-click (Windows) or Command-click (Macintosh) each screen.

Add media content to screens

- ❖ Add media content to the screen that is currently selected in the Screen Outline pane.

Edit an item on a screen

- ❖ Select the item in the Document window.

View the contents of a child screen when the parent screen appears

- ❖ To turn off the Hide feature (Hide Screen is selected for slide screens by default.), click Hide Screen in the child screen's context menu.

Select a child screen on the Stage

- 1 Deselect Hide Screen.
- 2 Select the parent screen in the Screen Outline pane.
- 3 Click in the contents of the child screen on the Stage.

Edit an item on an ancestor screen of the current screen

- ❖ Double-click the item in the Document window.

Note: By default, items on ancestor screens of the current screen are dimmed in the Document window.

Fully render all items on ancestor screens

- ❖ Select View > Preview Mode > Full.

Cut or copy a screen

- ❖ Do any of the following:
 - Right-click (Windows) or Control-click (Macintosh) the screen, and select Cut or Copy from the context menu.
 - Select Edit > Cut or Edit > Copy.

Paste a screen

- ❖ Do either of the following:
 - After cutting or copying the screen, right-click (Windows) or Control-click (Macintosh) another screen and select Paste from the context menu. The cut or copied screen is pasted after the selected screen. To nest the pasted screen within the selected screen, select Paste Nested Screen from the context menu.
 - After cutting or copying the screen, select Edit > Paste In Center or Edit > Paste In Place.

Drag a screen in the Screen Outline pane

- ❖ Using the mouse, drag the screen to any other position in the Screen Outline pane. Release the mouse button when the screen is in the desired position. To nest a screen within another screen, drag it towards the right side of the Screen Outline pane below the intended parent.

Remove a screen

- ❖ Do one of the following:
 - Right-click (Windows) or Control-click (Macintosh) the screen, and select Cut or Delete from the context menu.
 - Select the screen, and click the Delete Screen (-) button at the top of the Screen Outline pane.
 - Press Backspace (Windows) or Delete (Macintosh).

About screen names

By default, screens are named with their default type, in the order in which they are created: slide1, slide2, form1, form2, and so on. The creation order does not necessarily reflect the order of the screens in the Screen Outline pane. For example, you could create three sibling screens, slide1, slide2, and slide3. If you then create a nested screen directly below slide1, the nested screen is slide4.

To rename a screen, including the top-level screen, double-click the screen name in the Screen Outline pane and enter the new name. Screen names must be unique in a document. For example, a document can have only one screen named *Overview_Page*.

The default screen name is used as the instance name, which is used in ActionScript to control a screen. If you change the default screen name, the instance name is updated with the new name; likewise, if you change the instance name, the screen name is updated. The linkage identifier for the screen is also identical to the screen name, and it is updated when the screen name or instance is updated.

Instance names must conform to the following requirements:

- The name must not contain any spaces.
- The first character must be a letter, underscore (_), or dollar sign (\$).
- Each subsequent character must be a letter, number, underscore, or dollar sign.
- The instance name must be unique.

You can also change the instance name in the Property inspector.

To learn more about creating screen-based documents that use ActionScript, see “About organizing code for screens in” in *Using ActionScript 2.0 Components*.

See also

“[Set properties and parameters for a screen](#)” on page 323

Undoing and redoing commands with screens

To undo and redo the following actions performed on screens, use the Edit > Undo and Edit > Redo menu commands: adding, cutting, copying, pasting, deleting, and hiding a screen. The following actions performed on screens are recorded in the History panel: adding a screen, adding a nested screen, selecting a screen, renaming a screen, and deleting a screen.

See also

[“Undo, Redo, and Repeat commands”](#) on page 29

Use Find And Replace in a document with screens

You can search for a text string, font, color, symbol, sound file, video file, or imported bitmap file.

You can search for elements in the entire document or in the current screen.

- 1 Select Edit > Find And Replace.
- 2 Do one of the following:
 - To search the entire document, select Current Document from the Search In pop-up menu.
 - To search a screen, click in the Screen Outline pane, and select Current Screen from the Search In pop-up menu.

See also

[“Find and Replace”](#) on page 58

Accessibility in the Flash screens authoring environment

Using keyboard shortcuts rather than the mouse, users can navigate a document and use interface elements, including screens, panels, the Property inspector, dialog boxes, the Stage, and objects on the Stage.

Accessibility support for screen-based documents is similar to support for other documents, except that when keyboard shortcuts are used to navigate panels (Control+F6 in Windows or Command+F6 in Mac OS), the Screen Outline pane receives focus the first time the keyboard shortcut is used. For other documents, the Timeline receives focus first).

The Screen Outline pane receives focus only the first time you cycle through the panels. That is, if you arrive at the last panel and press the keyboard shortcut again, the Screen Outline pane is skipped, and the next panel receives focus.

To cycle through individual screens in the Screen Outline pane, you use the arrow keys.

See also

[“Creating accessible content”](#) on page 302

Adding content to screens

Set properties and parameters for a screen

To set properties and parameters for individual screens, use the Property inspector. On the left side of the Property inspector, you can view the instance name, width, height, and *x* and *y* coordinates of a screen.

- The instance name is a unique name assigned to a screen, used when you target the screen in ActionScript. Each screen is assigned a default instance name, based on its default name in the Screen Outline pane. The instance name and default screen name are also identical to the linkage identifier for the screen. If you update the instance name, the default screen name and the linkage identifier are also updated.
- Width and height are specified in pixels. The values in the W and H fields are read-only. Width and height are determined by the screen contents. To make sure the registration point stays in the same relative position when the screen width and height change, use the Auto Snap option.

- The *x* and *y* coordinates of a screen are specified in pixels. Move a child screen on the Stage by changing its *x* and *y* coordinates. Change the registration point of a screen using the registration grid.

To control screen behavior during playback, set parameters for slide and form screens.

Change the instance name of a screen

- 1 Select a screen in the Screen Outline pane.
- 2 Select Window > Properties.
- 3 On the left side of the Property inspector, enter a name in the Instance Name box.

Note: If you update the instance name, the screen name in the Screen Outline pane and the linkage identifier for the screen also update.

Move a child screen on the Stage

- 1 Deselect Hide Screen for the child screen.
- 2 Select the screen's parent in the Screen Outline pane, and select the child screen's content on the Stage.
- 3 Select Window > Properties.
- 4 In the Property inspector, enter new values for the *x* and *y* coordinates, drag the child screen to another location on the Stage, or use the Align panel.

Specify the ActionScript class and registration point of a screen

Specify the ActionScript class of the screen and its registration point on the Properties tab of the Property inspector:

- The ActionScript class specifies what class the screen belongs to. The class determines what methods and properties are available for the screen. By default, slide screens are assigned to the `mx.screens.Slide` class, and form screens are assigned to the `mx.screens.Form` class. You can assign the screen to a different class.
- The registration grid indicates the position of the screen registration point in relation to its content. By default, the registration point of a slide screen is in the center and Auto Snap is on. By default, the registration point of a form screen is in the upper left corner and Auto Snap is off. To change the registration point, use the grid. To keep the registration point in the same position in relation to screen contents, even when you add, remove, or reposition the screen contents, use the Auto Snap option.

The height and width of a screen are determined by its content. Therefore, the center of a screen cannot be the center of the Stage.

Note: If you changed the coordinate grid setting in the Info panel in another Flash document, the coordinate grid for the screen registration point can reflect that change. To check the Info panel coordinate grid setting, open a non-screen-based Flash document or select something on the Stage that is not a screen, and select Window > Info. To change settings in the Info panel while working in a screen-based document, deselect all screens before you open the panel.

See also

[“Get information about instances on the Stage”](#) on page 156

Change the ActionScript class of a screen

- 1 Select a screen in the Screen Outline pane.
- 2 Select Window > Properties.
- 3 In the Property inspector, click the Properties tab.

- 4 Enter a class name in the Class Name box. For more information on ActionScript classes, see Classes in [Learning ActionScript 2.0 in Adobe Flash](#).

Change the registration point of a screen

- 1 Select a screen in the Screen Outline pane.
- 2 Select Window > Properties.
- 3 Click the Properties tab and click a point in the registration grid.

Clicking a different registration point in the registration grid changes the point on the screen that is used as the registration point. When this is done, the registration point moves in relation to the screen content, but the screen itself does not move.

Setting parameters for a screen

To control how the screen appears and behaves during playback, set parameters on the Parameters tab of the Property inspector. Different parameters are available for slide and form screens.

The following parameters are available only for slide screens:

- The `autoKeyNav` parameter determines whether the slide uses default keyboard handling to control navigation to the next or previous slide. When `autoKeyNav` is set to `true`, pressing the Right Arrow key or the Spacebar advances to the next slide, and pressing the Left Arrow key moves to the previous slide. When `autoKeyNav` is set to `false`, no default keyboard handling takes place. When `autoKeyNav` is set to `inherit` (the default setting), the slide inherits its `autoKeyNav` setting from its parent. If the slide's parent is also set to `inherit`, the parent's ancestors are examined until one is found with its `autoKeyNav` parameter set to `true` or `false`. If a slide is a root slide, setting `autoKeyNav` to `inherit` yields the same result as setting it to `true`.

Note: This property can be set independently for each slide, and it affects keyboard handling when that slide has focus.

- The `overlayChildren` parameter specifies whether child screens overlay one another on the parent screen during playback. When `overlayChildren` is set to `true`, child screens overlay one another. For example, suppose you have two children, Child 1 and Child 2, which are bullet points on the parent screen. If the user clicks a Next button and displays Child 1, then clicks Next again and displays Child 2, Child 1 remains visible when Child 2 appears. When `overlayChildren` is set to `false` (the default setting), Child 1 is removed from the display when Child 2 appears. This parameter affects only the immediate children of a slide, not nested descendants.
- The `playHidden` parameter specifies whether a slide continues to play if it is hidden after being shown. When `playHidden` is set to `true` (the default setting), the slide continues to play when the slide is hidden after being shown. When `playHidden` is set to `false`, the slide stops playing if it is hidden, and resumes playing at Frame 1 if it is shown again.

One parameter is available only for form screens: The `visible` parameter indicates whether a screen is visible or hidden at runtime. When `visible` is set to `true`, the screen is visible at runtime. When `visible` is set to `false`, the screen is hidden. This property does not affect the visibility of the screen in the authoring environment.

The following parameters are available for slide and form screens:

- The `autoload` parameter indicates whether the content should load automatically (`true`), or wait to load until the `Loader.load()` method is called (`false`). The default value is `true`. This parameter is inherited from the Loader component.

- The `contentPath` parameter is an absolute or relative URL indicating the file to load when the `Loader.load()` method is called. A relative path must point to the SWF file loading the content. The URL must be in the same subdomain as the URL where the Flash content currently resides. For use in Flash Player or with the Test Movie command, all SWF files must be stored in the same folder, and the filenames cannot include folder or disk-drive specifications. The default value is undefined until the load starts. This parameter is inherited from the Loader component.

Specify parameter settings for a screen

- 1 Select a screen in the Screen Outline pane.
- 2 Select Window > Properties.
- 3 In the Property inspector, click the Parameters tab.
- 4 Click the setting for a parameter, and select a setting from the pop-up menu.

Creating controls and transitions for screens with behaviors

To create controls and transitions for screens, use behaviors. Controls enable the flow between screens—for example, you can go to another screen, hide a screen, or show a screen. Transitions create visual animations that play as the Flash document display changes from one screen to another.

Behaviors are built-in ActionScript scripts that you add to an object, such as a screen, to control that object. Behaviors let you add the power, control, and flexibility of ActionScript coding to your document without having to create the ActionScript code yourself. Behaviors are available for a variety of objects in Flash, including movie clips, text fields, and video and sound files.

To learn more about creating screen-based documents that use ActionScript, see "About organizing code for screens" in [Using ActionScript 2.0 Components](#).

Add controls to screens using behaviors

To add a control to a screen using a behavior, attach the behavior to a trigger—a button, movie clip, or screen—and target the screen to affect with the behavior. Select the event that triggers the behavior.

To control slide screens, add the following behaviors: Go to First Slide, Go to Last Slide, Go to Next Slide, Go to Previous Slide, and Go to Slide (specify slide name).

Note: *Go to Next Slide and Go to Previous Slide move to screens on the same level, not to parents or children.*

To control slide or form screens, add the following behaviors: Show a Specified Screen (if the screen was previously hidden) or Hide a Specified Screen (if the screen was previously shown).

- 1 Select the button, movie clip, or screen that you want to trigger the behavior.
- 2 In the Behaviors panel, click the Add (+) button.
- 3 Select Screen, and select the desired control behavior from the submenu.
- 4 If the behavior requires that you select a target screen, the Select Screen dialog box appears. Select the target screen in the tree control. Click Relative to use a relative target path, or Absolute to use an absolute target path, and click OK.

Note: *Some behaviors select a target screen by default; for example, the Go to First Slide screen automatically targets the first screen. These behaviors do not show the Select Screen dialog box.*

- 5 In the Event column, click in the row for the new behavior and select an event from the list. This specifies the event that triggers the behavior—for example, a user clicking a button, a movie clip loading, or a screen receiving focus. The list of available events depends on the type of object you use to trigger the behavior.

See also

[“Document structure and hierarchy”](#) on page 317

[“Using absolute and relative target paths”](#) on page 181

Adding transitions to screens using behaviors

Screen transition behaviors let you add animated transitions between screens. To add a transition using a behavior, you attach the behavior directly to a screen.

You can select the direction of a transition: In plays the animation as the screen first appears in the document; Out plays the animation as the screen disappears from the document. You can also specify the duration in seconds.

Easing options let you modify the transition to achieve different effects.

Some transitions have additional parameters that you can modify. Parameters appear in the Transitions dialog box when you select the transition.

Follow these guidelines when adding transitions:

- For most situations, the In option is recommended.
- Use the In option when applying a transition that uses the `on(reveal)` event.
- Use the Out option when applying a transition that uses the `on(hide)` event.
- Do not add an Out transition immediately before an In transition in a presentation.
- To attach the same transition to all children of a given slide, attach the single transition to the `on(revealChild)` or `on(hideChild)` event of the parent, rather than duplicating the transition on all child slides.

Add a transition behavior

- 1 Select the screen to apply the behavior to.
- 2 In the Behaviors panel, click the Add (+) button.
- 3 Select Screen > Transition from the submenu.
- 4 Select a transition from the scroll list. An animated preview of the transition plays in the preview window, and a brief description of the transition appears in the description field. The animation changes to reflect options that you select for the transition in the following steps.
- 5 For Direction, select In to play the transition as the screen appears in the document, and Out to play the transition as the screen disappears from the document.
- 6 For Duration, enter a time in seconds.
- 7 For Easing, select an option to define the transition style.
- 8 If the transition has additional parameters, select options or enter values for those parameters in the fields provided.
- 9 Click OK.
- 10 In the Behaviors panel, in the Event column, click in the row for the new behavior and select an event from the list. This action specifies the event that triggers the behavior—for example, the mouse pointer moving over the screen.

Screen instance names, class names, and registration points

The screen name automatically generates the instance name and class name of the screen. You need these identifying labels when you manipulate screens with ActionScript in various ways. To adjust how the screen behaves, change a screen's registration point. You can work with these features in the following ways:

- The instance name is a unique name assigned to a screen, used when you target the screen in ActionScript. Change the instance name in the Property inspector. The instance name is identical to the screen name in the Screen Outline pane and the linkage identifier for the screen. If you update the instance name, the screen name and the linkage identifier also update.

Note: Symbol instances, including movie clips, buttons, and graphics, also have instance names.

- The class name identifies the ActionScript class to which the screen is assigned. By default, a slide screen is assigned to the `mx.screens.Slide` class, and a form screen is assigned to the `mx.screens.Form` class. To modify the methods and properties that are available for the screen, assign the screen to a different class. For more information on ActionScript classes, see Classes [Learning ActionScript 2.0 in Adobe Flash](#).
- The Property inspector indicates the registration point in the *x* and *y* coordinate fields and in the registration grid. You might want to move the registration point for greater control in manipulating screen content. For example, to create a spinning shape in the center of a screen, reposition the screen registration point at the center of the screen and rotate the screen around its registration point.

See also

[“Symbols, instances, and library assets”](#) on page 147

How screens interact with ActionScript

Screens are similar to nested movie clips in the way that they interact with ActionScript. However, some differences exist.

Use the following guidelines for ActionScript with screens:

- When you select a screen in the Screen Outline pane and add ActionScript, the script is added directly to the screen as an object action (much as ActionScript is added directly to a movie clip). Use object actions for simple code (such as creating navigation between screens) and external ActionScript files for more complex code.
- For best results, organize the document structure and finalize screen names before adding ActionScript. If you rename a screen, the instance name is automatically changed, and you must update the instance names in any ActionScript code you write.
- To add a frame action to the Timeline for a screen, select the screen, open the Timeline (Window > Timeline), and select the first frame in the Timeline. Use an external ActionScript file, rather than a frame action, for complex code on a screen.
- You cannot view or manipulate the main Timeline for a screen-based document. However, you can target the main Timeline using `_root` in a target path.
- Each screen is automatically associated with ActionScript, based on its class. You can change the class to which that screen is assigned, and you can set some parameters for a screen in the Property inspector.
- To control screens with ActionScript, use the Screen class, Slide class, and Form class.
- To create interactivity, use components whenever possible. Put no more than 125 total component instances in a single FLA file.
- To create navigation between slides, use `rootSlide`. For example, to get the current slide, use `rootSlide.currentSlide`.

- Do not try to do slide navigation inside of `on(reveal)` or `on(hide)` handlers.
- Do not add an `on(keydown)` or `on(keyup)` event to ActionScript code controlling a screen.

For more information on controlling screens with ActionScript, see “Screen class”, “Form class”, and “Slide class”, in the *ActionScript 2.0 Components Language Reference*.

For information on the Object class and the `onClipEvent()` event handler, see Object and `onClipEvent` handler in *ActionScript 2.0 Language Reference*.

To learn more about creating screen-based documents that use ActionScript, see “About organizing code for screens” in *Using ActionScript 2.0 Components*.

See also

“[About nested movie clips and parent-child hierarchy](#)” on page 178

Using components with screens

To create complex, structured applications in Flash, use components with screens. Components are especially useful with forms to create structured applications that show data and enable nonlinear user interactivity. For example, use forms to populate a container component.

To create custom navigation between components, use the Focus Manager when you use components with screens. The Focus Manager specifies the order in which components receive focus when a user presses the Tab key to navigate in an application. For example, customize a form application so that a user can press Tab to navigate fields and press Return (Macintosh) or Enter (Windows) to submit the form.

For information on the Focus Manager, see “Creating custom focus navigation” in *Using ActionScript 2.0 Components* and “FocusManager class” in *ActionScript 2.0 Components Language Reference*.

You can also use the Accessibility panel to create a tab order.

See also

“[Viewing and creating tab order and reading order](#)” on page 309

Chapter 15: ActionScript

ActionScript® is the scripting language in Flash. Use ActionScript to make your applications play in a nonlinear way, and to add interesting or complex functionality that cannot be represented in the Timeline.

Working with ActionScript

About ActionScript

The ActionScript® scripting language lets you add complex interactivity, playback control, and data display to your application. You can add ActionScript in the authoring environment by using the Actions panel, Script window, or an external editor.

ActionScript follows its own rules of syntax, reserved keywords, and lets you use variables to store and retrieve information. ActionScript includes a large library of built-in classes that let you create objects to perform many useful tasks. For more information on ActionScript, see the following titles:

- *Programming ActionScript 3.0* at www.adobe.com/go/learn_fl_cs4_programmingAS3_en
- *ActionScript 3.0 Language and Components Reference* at www.adobe.com/go/learn_flashcs4_langref_en
- *Learning ActionScript 2.0 in Adobe Flash* at www.adobe.com/go/learn_fl_cs4_learningAS2_en
- *ActionScript 2.0 Language Reference* at www.adobe.com/go/learn_flcs4_as2lr_en

You don't need to understand every ActionScript element to begin scripting; if you have a clear goal, you can start building scripts with simple actions.

ActionScript and JavaScript are both rooted in the ECMA-262 standard, the international standard for the ECMAScript scripting language. For this reason, developers who are familiar with JavaScript should find ActionScript immediately familiar. For more information about ECMAScript, go to ecma-international.org.

Using the ActionScript documentation

Because there are multiple versions of ActionScript (2.0 and 3.0), and multiple ways of incorporating it into your FLA files, there are several different ways to learn ActionScript.

This help system describes the graphical user interface for working with ActionScript. This interface includes the Actions panel, Script window, Script Assist mode, Behaviors panel, Output panel, and Compiler Errors panel. These topics apply to all versions of ActionScript.

Other ActionScript documentation from Adobe will help you learn about the individual versions of ActionScript; see *Programming ActionScript 3.0*, *Learning ActionScript 2.0 in Adobe Flash*, *Developing Flash Lite 1.x Applications* or *Developing Flash Lite 2.x Applications*. For information about the ActionScript vocabulary, see the *ActionScript Language Reference* for the version you are working with.

For video tutorials about ActionScript 3.0, the Flash workflow, and components, see the following:

- Getting started with ActionScript 3.0: www.adobe.com/go/vid0129
- Creating interactivity with ActionScript 3.0: www.adobe.com/go/vid0130
- Flash workflow: www.adobe.com/go/vid0132
- Using components: www.adobe.com/go/vid0133

For text tutorials about ActionScript, see www.adobe.com/go/learn_fl_tutorials. The following tutorials are available:

- Create an Application
- Add Interactivity
- Work with Objects and Classes

ActionScript versions

Flash includes more than one version of ActionScript to meet the needs of different kinds of developers and playback hardware.

- ActionScript 3.0 executes extremely fast. This version requires somewhat more familiarity with object-oriented programming concepts than the other ActionScript versions. ActionScript 3.0 is fully compliant with the ECMAScript specification, offers better XML processing, an improved event model, and an improved architecture for working with onscreen elements. FLA files that use ActionScript 3.0 cannot include earlier versions of ActionScript.
- ActionScript 2.0 is simpler to learn than ActionScript 3.0. Although Flash Player runs compiled ActionScript 2.0 code slower than compiled ActionScript 3.0 code, ActionScript 2.0 is still useful for many kinds of projects that are not computationally intensive; for example, more design-oriented content. ActionScript 2.0 is also based on the ECMAScript spec, but is not fully compliant.
- ActionScript 1.0 is the simplest form of ActionScript, and is still used by some versions of the Flash Lite Player. ActionScript 1.0 and 2.0 can coexist in the same FLA file.
- Flash Lite 2.x ActionScript is a subset of ActionScript 2.0 that is supported by Flash Lite 2.x running on mobile phones and devices.
- Flash Lite 1.x ActionScript is a subset of ActionScript 1.0 that is supported by Flash Lite 1.x running on mobile phones and devices.

Ways of working with ActionScript

There are several ways to work with ActionScript.

- Script Assist mode lets you add ActionScript to your FLA file without writing the code yourself. You select actions, and the software presents you with a user-interface for entering the parameters required for each one. You must know a little about what functions to use to accomplish specific tasks, but you don't have to learn syntax. Many designers and non-programmers use this mode.
- Behaviors also let you add code to your file without writing it yourself. Behaviors are prewritten scripts for common tasks. You can add a behavior and then easily configure it in the Behaviors panel. Behaviors are available only for ActionScript 2.0 and earlier.
- Writing your own ActionScript gives you the greatest flexibility and control over your document, but it requires you to become familiar with the ActionScript language and conventions.
- Components are prebuilt movie clips that help you implement complex functionality. A component can be a simple user interface control, such as a check box, or it can be a complicated control, such as a scroll pane. You can customize a component's functionality and appearance, and you can download components created by other developers. Most components require you to write some ActionScript code of your own to trigger or control a component. For more information, see [About ActionScript 3.0 Components](#) in *Using ActionScript 3.0 Components* or [About components](#) in *Using ActionScript 2.0 Components* at www.adobe.com/go/learn_fl_cs4_as2components_en.

Writing ActionScript

When you write ActionScript code in the authoring environment, you use the Actions panel or Script window. The Actions panel and Script window contain a full-featured code editor that includes code hinting and coloring, code formatting, syntax highlighting, syntax checking, debugging, line numbers, word wrapping, and support for Unicode.

- Use the Actions panel to write scripts that are part of your Flash document (that is, scripts that are embedded in the FLA file). The Actions panel provides features such as the Actions toolbox, which gives you quick access to the core ActionScript language elements, and Script Assist mode, in which you are prompted for the elements needed to create scripts.
- Use the Script window if you want to write external scripts—that is, scripts or classes that are stored in external files. (You can also use a text editor to create an external AS file.) The Script window includes code-assistance features such as code hinting and coloring, syntax checking, and auto-formatting.

See also

[“Symbols and ActionScript”](#) on page 168

[“Timelines and ActionScript”](#) on page 180

[“Sound and ActionScript”](#) on page 280

[“Controlling external video playback with ActionScript”](#) on page 296

[“Multilanguage text and ActionScript”](#) on page 270

[“Creating accessibility with ActionScript”](#) on page 312

[“Organizing ActionScript in an application”](#) on page 420

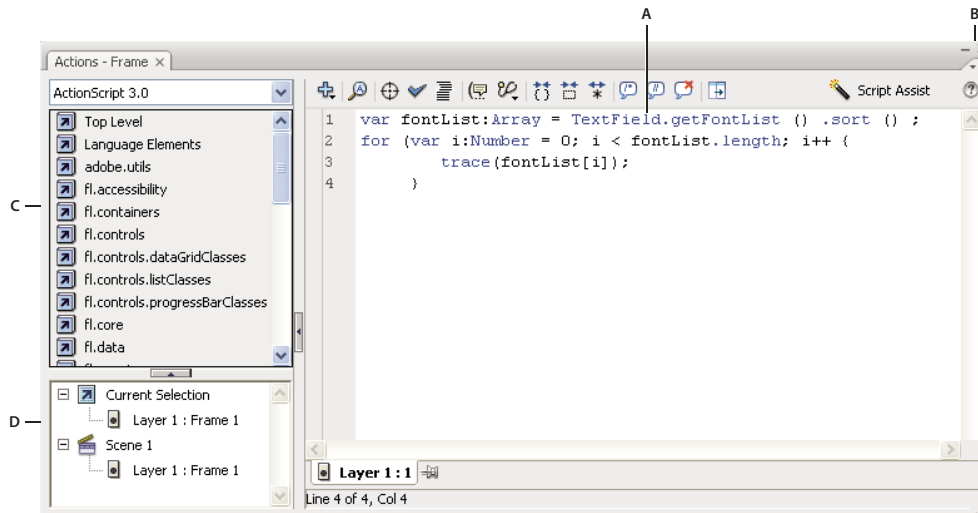
[“Debugging ActionScript 1.0 and 2.0”](#) on page 350

[“Debugging ActionScript 3.0”](#) on page 360

[“Script Assist mode and behaviors”](#) on page 337

Actions panel overview

To create scripts embedded in a FLA file, enter ActionScript directly into the Actions panel. The Actions panel consists of three panes: the Actions toolbox, which groups ActionScript elements by category; the Script navigator, which lets you move quickly between the scripts in your Flash document; and the Script pane, where you type your ActionScript code.



A. Script pane B. Panel menu C. Actions toolbox D. Script navigator

See also

“Pin scripts in the Actions panel” on page 345

Display the Actions panel

- ❖ Select Window > Actions or press F9.

Use the Actions toolbox


- ❖ Insert an ActionScript element into the Script pane by double-clicking or dragging it directly into the Script pane.

The Actions toolbox separates items into categories, and also provides an alphabetical index.

Use the Script pane

- ❖ Type your code.

Resize the Actions toolbox or Script pane

- Drag the vertical bar that appears between the Actions toolbox and Script pane.
- Click the Show/Hide Toolbox button  at the top of the Actions panel.
- Double-click the vertical bar to collapse the Actions toolbox; double-click the bar again to display the Actions toolbox.
- Click the arrow on the vertical bar to expand or collapse the Actions toolbox.

When the Actions toolbox is hidden, you can still use the Add (+) button to access its items.

Use the Script navigator

- Click an item in the Script navigator; the script associated with that item appears in the Script pane, and the playhead moves to that position on the timeline.
- Double-click an item in the Script navigator to *pin* the script (lock it in place).

Print actions

- 1 From the Actions panel pop-up menu, select Print.

2 Select Options and click Print.


Because the printed copy won't include information about file it came from, you should include information such as the name of the FLA file in a comment action in the script.

Access context-sensitive Help from the Actions panel

1 To select an item for reference, do any of the following:

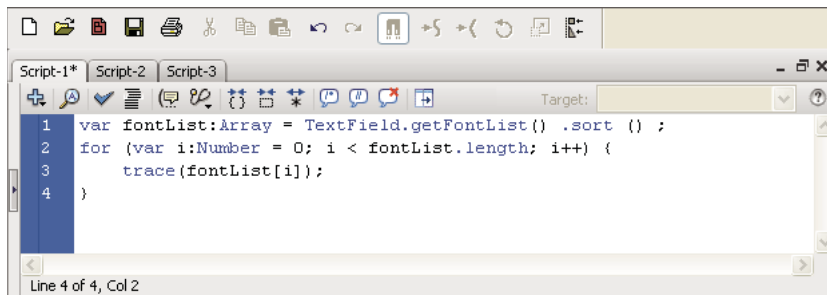
- Select an ActionScript term in the Actions panel toolbox pane (on the left side of the Actions panel).
- Select an ActionScript term in the Actions panel in the Script pane.
- Place the insertion point before an ActionScript term in the Actions panel in the Script pane.

2 To open the Help panel reference page for the selected item, do one of the following:

- Press F1.
- Right-click the item and select View Help.
- Click Help  above the Script pane.

Script window overview

The Script window lets you create external script files that you import into your application. These scripts can be ActionScript, Flash Communication, or Flash JavaScript files. The Add (+) menu lists the language elements available for the type of script you create.



Script window

If you have more than one external file open, filenames are displayed on tabs across the top of the Script window.

In the Script window, you can use the following features: the Add (+) menu (which is like the Actions toolbox), find and replace, syntax checking, syntax coloring, auto format, code hinting, code commenting, code collapse, debug options (ActionScript files only), and word wrap. The Script window also lets you display line numbers and hidden characters.

The Script window does not include code-assistance features such as the Script navigator, Script Assist mode, and behaviors. These features are useful only in the context of creating a FLA file, not an external script file.

See also

“[Writing and managing scripts](#)” on page 339

Create an external file in the Script window

1 Select File > New.


- 2 Select the type of external file you want to create (ActionScript file, ActionScript Communication file, or Flash JavaScript file).


Edit an existing file in the Script window


- To open an existing script, select File > Open, and then open an existing AS file.
- To edit a script that is already open, click the document tab that shows the script's name.


Tools in the Actions panel and Script window


The Actions panel and Script window toolbars let you access the code-assistance features that help simplify and streamline coding in ActionScript. The tools are different depending on whether you are using the Actions panel or the Script window.


Add A New Item To The Script  Displays the language elements that are also in the Actions toolbox. Select an item to add it to the script.


Find  Finds and replaces text in your script.


Insert Target Path  (Actions panel only) Helps you set an absolute or relative target path for an action in the script.


Check Syntax  Checks for syntax errors in the current script. Syntax errors are listed in the Output panel.


Auto Format  Formats your script for proper coding syntax and improved readability. Set autoformatting preferences in the Preferences dialog box, which is available from the Edit menu or from the Actions Panel menu.


Show Code Hint  If you've turned off automatic code hinting, use Show Code Hint to display a code hint for the line of code you're working on.


Debug Options  (Actions panel only) Sets and removes breakpoints so that you can proceed line by line through your script when debugging. You can use debug options only for ActionScript files, not ActionScript Communication or Flash JavaScript files.


Collapse Between Braces  Collapses code that appears between the curly braces or parentheses that currently contain the insertion point.


Collapse Selection  Collapses the currently selected code block.


Expand All  Expands all collapsed code in the current script.


Apply Block Comment  Adds comment markers to the beginning and end of the selected code block.


Apply Line Comment  Adds a single-line comment marker at the insertion point, or at the beginning of each line of code in a multiline selection.

Remove Comment  Removes comment markers from the current line or all lines of the current selection.

Show/Hide Toolbox  Displays or hides the Actions Toolbox.

Script Assist  (Actions panel only) In Script Assist mode, you are presented with a user-interface for entering the elements needed to create scripts.

Help  Displays reference information for the ActionScript element that is selected in the Script pane. For example, if you click an `import` statement and then click Help, the reference information for `import` appears in the Help panel.

Panel menu  (Actions panel only) Contains the commands and preferences that apply to the Actions panel. For example, you can set line numbers and word wrapping, access ActionScript preferences, and import or export scripts.

See also

[“Writing and managing scripts”](#) on page 339

[“Debugging ActionScript 3.0”](#) on page 360

[“Debugging ActionScript 1.0 and 2.0”](#) on page 350

Set ActionScript preferences

Whether you edit code in the Actions panel or the Script window, you can set and modify a single set of preferences.

1 Select Edit > Preferences (Windows) or Flash > Preferences (Macintosh), and then click ActionScript in the Category list.

2 Set any of these preferences:

Automatic Indentation When automatic indentation is turned on, the text you type after an opening parenthesis “(“ or opening curly brace “{“ is automatically indented according to the Tab Size setting.

Tab Size Specifies the number of characters a new line is indented.

Code Hints Enables code hinting in the Script pane.

Delay Specifies the delay (in seconds) before code hints are displayed.

Font Specifies the font used for your script.

Use Dynamic Font Mapping Checks to ensure that the selected font family can render each character. If not, Flash substitutes a font family that contains the necessary characters.

Open/Import Specifies the character encoding used when you open or import ActionScript files.

Save/Export Specifies the character encoding used when you save or export ActionScript files.

Reload Modified Files Specifies what happens when a script file is modified, moved, or deleted. Select Always, Never, or Prompt.

- **Always** No warning is displayed, and the file is automatically reloaded.
- **Never** No warning is displayed, and the file remains in the current state.
- **Prompt** (Default) A warning is displayed, and you can choose whether to reload the file.

When you build applications with external scripts, this preference helps you avoid overwriting a script that a team member has modified since you opened the application, or avoid publishing the application with older versions of scripts. The warnings let you automatically close a script and reopen the newer, modified version.

Syntax Colors Specifies code coloring in your scripts.

Language Opens the ActionScript Settings dialog boxes, where you can set a classpath for ActionScript 2.0, or a source path, library path, external library path for ActionScript 3.0.

See also

[“Set the source path for ActionScript 3.0”](#) on page 374

[“Set the classpath for ActionScript 2.0”](#) on page 374

[“Import and export scripts”](#) on page 344

[“Format code”](#) on page 339

[“Use code hints”](#) on page 347

Script Assist mode and behaviors

About Script Assist mode

If you are new to ActionScript, or if you want to add simple interactivity without having to learn the ActionScript language and its syntax, you can use Script Assist in the Actions panel to help you add ActionScript to your FLA files.

Script Assist lets you build scripts by selecting items from the Actions toolbox. When you click an item once, its description appears at the upper right of the panel. When you double-click an item, it adds the item to the Actions panel Script pane.

In Script Assist mode, you can add, delete, or change the order of statements in the Script pane; enter parameters for actions in boxes above the Script pane; find and replace text; and view script line numbers. You can also *pin* a script—that is, keep a script in the Script pane when you click away from the object or frame.

Script Assist helps you avoid the syntax and logic errors a novice user might make. However, to use Script Assist you must become familiar with ActionScript, and know what methods, functions, and variables to use when creating your scripts. To learn about ActionScript, see *Learning ActionScript 2.0 in Adobe Flash* or *Programming ActionScript 3.0*.

For a video tutorial about Script Assist mode, see www.adobe.com/go/vid0131.

For a text tutorial about Script Assist mode, see Use Script Assist Mode on the Flash Tutorials page at www.adobe.com/go/learn_fl_tutorials.


Use Script Assist to write ActionScript

To add an ActionScript 3.0 action to a Flash document, you must attach it to a frame. To add an ActionScript 2.0 (or earlier) action to a Flash document, attach it to a button or movie clip, or to a frame in the timeline.







For a video tutorial about Script Assist mode, see www.adobe.com/go/vid0131.

For a text tutorial about Script Assist mode, see Use Script Assist Mode on the Flash Tutorials page at www.adobe.com/go/learn_fl_tutorials.

Start Script Assist mode

- 1 Select Window > Actions.
- 2 In the Actions panel, click Script Assist .

In Script Assist mode, the Actions panel changes in the following ways:

- Add (+) functions differently in Script Assist mode. When you select an item from the Actions toolbox or the Add menu , the item is added after the currently selected text block.
- Delete (-) lets you remove the current selection in the Script pane.
- The up and down arrows let you move the current selection in the Script pane upward or downward within the code.
- The Check Syntax , Auto Format , Show Code Hint , and Debug Options  buttons and menu items normally visible in the Actions panel are disabled, as they do not apply to Script Assist mode.
- The Insert Target button  is disabled unless you are typing in a box. Clicking Insert Target places the resulting code in the current box.

Note: If the Actions panel contains ActionScript code when you click Script Assist, Flash compiles the code. If there are errors in the code, you cannot use Script Assist until you fix the current code selection. The errors are described in detail in the Compiler Errors panel.

View a description of an action

- Click a category in the Actions toolbox to display the actions in that category, and click an action.
- Select a line of code in the Script pane.

The description appears at the top of the Actions panel.

Add an action to the Script pane

- Click a category in the Actions toolbox to display the actions in that category, and then either double-click an action or drag it to the Script pane.
- Click Add (+) and select an action from the pop-up menu.
- Press Escape and a shortcut key. (To view a list of shortcut keys, select Esc Shortcut Keys in the Actions panel pop-up menu; select this option again to hide the list.)

Delete an action

- 1 Select a statement in the Script pane.
- 2 Click Delete (-) or press the Delete key.

Move a statement up or down in the Script pane

- 1 Select a statement in the Script pane.
- 2 Click the up or down arrow.


Work with parameters

- 1 Add an action to, or select a statement in, the Script pane.

Relevant parameter options appear above the Script pane.

- 2 Enter values in the boxes above the Script pane.

Search for text in a script

- To go to a specific line in a script, choose Go To Line from the Actions panel pop-up menu or press Control+G (Windows) or Command+G (Macintosh); then enter the line number.
- To find text, click Find , select Find from the Actions panel pop-up menu, or press Control+F (Windows) or Command+F (Macintosh).
- To find text again, press F3 or select Find Again from the Actions panel pop-up menu.
- To replace text, click Find, or press Control+H (Windows) or Command+H (Macintosh).

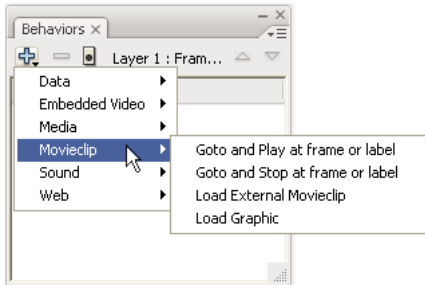
In Script Assist mode, Replace searches and replaces text only in the parameter box of each action, not in the entire script. For example, in Script Assist mode you cannot replace all `gotoAndPlay` actions with `gotoAndStop`.

About behaviors

Behaviors are predefined scripts that you can attach to objects in your FLA file. Behaviors provide functionality such as frame navigation, loading external SWF and JPEG files, controlling the stacking order of movie clips, and movie clip dragging.

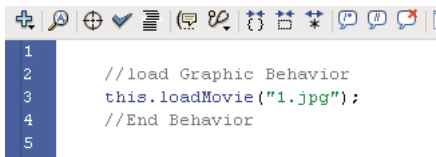
Behaviors provide a convenient way to avoid writing ActionScript and can help you learn how ActionScript works.

Behaviors are available only for ActionScript 2.0 and earlier, and only when you work in the Actions panel, not in an external script file. Typically, you select a triggering object in your document (such as a movie clip or a button), select Add in the Behaviors panel, and then select behavior, as the following example shows:



Selecting a behavior from the Behaviors panel

The behavior is added to the object and is displayed in the Actions panel.



The ActionScript of a behavior

For a sample about behaviors, see the Flash Samples page at www.adobe.com/go/learn_fl_samples. Download and decompress the Samples zip file and navigate to the Behaviors\BehaviorsScrapbook folder to access the sample.

See also

“Controlling instances with behaviors” on page 169

“Add and configure a behavior” on page 170


“Create custom behaviors” on page 170

Writing and managing scripts

Format code


Your code can be formatted and indented automatically or manually. If you use dynamic font mapping, you ensure that the correct fonts are used for multilingual text.

Set Auto Format options

- 1 Do one of the following:
 - From the panel menu  (at the upper-right corner of the Actions panel), select Preferences.
 - In the Script window, select Edit > Preferences (Windows) or Flash > Preferences (Macintosh).
- 2 In the Preferences dialog box, select Auto Format.
- 3 Select any of the Auto Format options.

After you set Auto Format options, your settings are applied automatically to the code you write, but not to existing code; you must apply your settings to existing code manually.

Format code according to Auto Format settings

- Click Auto Format  in the Actions panel or Script window toolbar.
- From the panel menu (at the upper-right corner of the Actions panel), select Auto Format.
- Press Control+Shift+F (Windows) or Command+Shift+F (Macintosh).
- In the Script window, select Tools > Auto Format.

Use dynamic font mapping

- ❖ To turn dynamic font mapping on or off, select or deselect Use Dynamic Font Mapping in the Preferences dialog box. Dynamic font mapping is turned off by default because it increases performance time when you are scripting. If you are working with multilingual text, turn on dynamic font mapping because it helps to ensure that the correct fonts are used.

Use automatic indentation

- ❖ To turn automatic indentation on or off, select or deselect Automatic Indentation in the Preferences dialog box.

When automatic indentation is turned on, the text you type after an opening parenthesis “(“ or opening curly brace “{“ is automatically indented according to the Tab Size setting in ActionScript preferences.


In your scripts, you can indent a line by selecting it and pressing Tab. To remove the indent, select the line and press Shift+Tab.

Comment sections of code

Code comments are parts of code that the ActionScript compiler ignores. A comment line explains what your code is doing, or temporarily deactivates code that you don't want to delete. Comment a line of code by beginning it with a double slash (//). The compiler ignores all text on that line after the double slash. You can also comment larger blocks of code by placing a slash and asterisk (/*) at the beginning of the block and an asterisk and slash (*/) at the end of the block.


You can type these comment markers manually, or you can use buttons at the top of the Actions panel or Script window to add them.

Comment a line of code

- 1 Place the insertion point at the beginning of the line or at the character where you want the comment to begin.
- 2 Click Apply Line Comment  at the top of the Actions panel or Script window.

A double slash (//) is placed at the insertion point.


Comment multiple lines of code

- 1 Select the lines to comment. (The first and last lines of the selection can be partial lines.)
- 2 Click Apply Block Comment  at the top of the Actions panel or Script window.

Block comment characters are placed at the beginning (/*) and end (*/) of the selection.

Remove a comment

- 1 Place the insertion point in the line that contains the comment, or select the block of code that is commented.


- 2 Click Remove Comment .

Use syntax coloring

In ActionScript, as in any language, *syntax* is the way elements are put together to create meaning. If you use incorrect ActionScript syntax, your scripts cannot work.

To highlight syntax errors, set a color-code for parts of your scripts. For example, suppose you set the syntax coloring preference to make keywords appear in blue. When you type `var`, the word `var` appears in blue. However, if you mistakenly type `vae`, the word `vae` remains black, which shows that you made a typing error.

Do one of the following:

- Select Edit > Preferences (Windows) or Flash > Preferences (Macintosh), click ActionScript in the Category list, and specify Syntax Coloring settings.
- From the panel menu  (at the upper-right corner of the Actions panel), select Preferences and specify Syntax Coloring settings.
- With the insertion point in the Script pane, press Control-U (Windows) or Command-U (Macintosh).

Note: When you write scripts in the Actions panel, commands that are not supported by the version of the player you are targeting appear in yellow in the Actions toolbox. For example, if the Flash Player SWF file version is set to Flash 7, ActionScript that only Flash Player 8 supports appears in yellow in the Actions toolbox.


See also

“Set ActionScript preferences” on page 336


Use line numbers and word wrap

When you edit or modify code, line numbers make code easier to scroll and parse. Word wrap helps you avoid horizontally scrolling long lines of code (especially when you work in the authoring environment, or at low screen resolutions).


Enable or disable line numbers

- From the panel menu  (at the upper-right corner of the Actions panel), select Line Numbers.
- In the Script window, select View > Line Numbers.
- Press Control+Shift+L (Windows) or Command+Shift+L (Macintosh).

Highlight a specific line

- From the panel menu  (at the upper-right corner of the Actions panel), select Go To Line.
- In the Script window, select Edit > Go To Line.


Enable or disable line word wrap

- From the panel menu  (at the upper-right corner of the Actions panel), select Word Wrap.
- In the Script window, select View > Word Wrap.
- Press Control+Shift+W (Windows) or Command+Shift+W (Macintosh).


Collapse sections of code

To make your code more readable and easier to navigate during programming and debugging, collapse sections of code into a single line. By collapsing sections that you don't need to look at, you can focus on the code you are writing or debugging.

Collapse selected code

- 1 Select the code to collapse.
- 2 Click Collapse Selection .


Collapse code between braces or parentheses

- 1 Place the insertion point inside the braces or parentheses.
- 2 Click Collapse Between Braces .

Expand collapsed code

- ❖ Click the plus sign (+) that appears to the left of the collapsed code. (To recollapse the code block, click the minus sign (-) that appears.)

Expand all collapsed code in the current script

- ❖ Click Expand All .

Adding ActionScript with shortcut keys

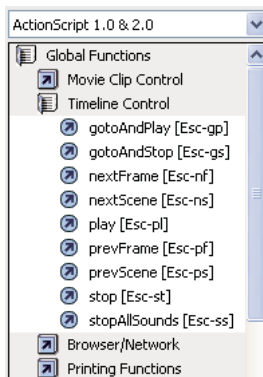
To add elements to a script, use Escape shortcut keys (pressing the Escape key, and then two other keys). For example, if you are working in the Script pane and press Escape+d+o, the following code is placed in your script:

```
do {
} while ();
```

The insertion point is placed after the word `while`, so you can begin typing your condition. Similarly, if you press Escape+c+h, the following code is placed in your script, and the insertion point is placed between the parentheses (), so you can begin typing your condition:

```
catch () {
}
```


To learn which commands have Escape shortcut keys, display them in the Actions toolbox by selecting Escape Shortcut Keys from the Actions Panel menu.




Escape shortcut keys

Display hidden characters

Characters such as spaces, tabs, and line breaks are hidden in ActionScript code. You may need to display these characters; for example, you must find and remove double-byte spaces that are not part of a string value, because they cause compiler errors.

- From the panel menu  (at the upper-right corner of the Actions panel), Select Hidden Characters.
- Press Control+Shift+8 (Windows) or Command+Shift+8 (Macintosh).

The following symbols are used to display hidden characters.

Hidden character	Symbol
Single-byte space	.
Double-byte space	
Tab	>>
line break	

Find text in a script


The Find tool lets you find and replace text strings in your scripts.

Note: To search through text in every script in a Flash document, use the Movie Explorer.


See also

“Use the Movie Explorer” on page 27


Find text

- 1 In the Actions panel or Script window, click Find  or press Control+F (Windows) or Command+F (Macintosh).
- 2 Enter the search string.
- 3 Click Find Next.

Find and replace text in a script

- 1 In the Actions panel or Script window, click Find  or press Control+F (Windows) or Command+F (Macintosh).
- 2 Enter the search string.
- 3 In the Replace box, enter the new string.
- 4 Click Find Next.
- 5 To replace the string, click Replace; to replace all occurrences of the string, click Replace All.

Repeat a search in the Actions panel

- ❖ From the panel menu  (at the upper-right corner of the Actions panel), select Find Again.

Repeat a search in the Script window

- ❖ Select Edit > Find Again.

Check syntax and punctuation



You can do a quick check of your ActionScript code without publishing the FLA file.

When you check syntax, the current script is checked. If the current script calls ActionScript classes, those classes are also checked. Other scripts that might be in the FLA file are not checked.

For ActionScript 2.0 files, Check Syntax runs the code through the compiler, generating syntax and compiler errors.

For ActionScript 3.0 files, Check Syntax generates only syntax errors. To generate compiler errors, such as type mismatches, improper return values, and variable or method name misspellings, you must use the Control > Test Movie command.

Check syntax

- In the Actions panel or Script window, click Check Syntax .
- From the panel menu  (at the upper-right corner of the Actions panel), select Check Syntax.
- Click in the Script pane, and then press Control+T (Windows) or Command+T (Macintosh).

Syntax errors are listed in the Compiler Errors panel.

***Note:** In an external ActionScript class file in the Script window, the global classpath (AS2) or source path (AS3) affects the syntax check. Even if the global classpath or source path is set correctly, you might generate errors, because the compiler is not aware that this class is being compiled. For more information about ActionScript 3.0 classes, see [Objects and classes](#) in *Programming ActionScript 3.0*. For more information on compiling ActionScript 2.0 classes, see *Compiling and exporting classes* in *Learning ActionScript 2.0 in Adobe Flash* at www.adobe.com/go/learn_fl_cs4_learningAS2_en.*

Check for punctuation balance

- 1 Click between braces {}, brackets [], or parentheses () in your script.
- 2 In Windows, press Control+' (single quote); on the Macintosh, press Command+' (single quote). The text between braces, brackets, or parentheses is highlighted, and you can check that opening punctuation has corresponding closing punctuation.

Import and export scripts

You can import a script into the Actions panel or Script window. You can also export your scripts from the Actions panel to external ActionScript files. (When you use the Script window, exporting is unnecessary because you can instead save the AS file.)

If text in your scripts doesn't look as expected when you open or import a file, change the import encoding preference.

Import an external AS file

- 1 In the Script pane, place the insertion point where you want to locate the first line of the external script.
- 2 Do one of the following:
 - In the Actions panel, select Import Script from the panel menu, or press Control+Shift+I (Windows) or Command+Shift+I (Macintosh).
 - In the Script window, select File > Import Script or press Control+Shift+I (Windows) or Command+Shift+I (Macintosh).

Export a script from the Actions panel

- 1 Select the script to export. Then select Export Script from the Actions Panel menu, or press Control+Shift+X (Windows) or Command+Shift+X (Macintosh).
- 2 Save the ActionScript (AS) file.

Set text encoding options

- 1 Select Edit > Preferences (Windows) or Flash > Preferences (Macintosh), and select ActionScript in the Category list.
- 2 Set any of these options:

Open/Import Select UTF-8 Encoding to open or import using Unicode encoding, or select Default Encoding to open or import using the encoding form of the language currently used by your system.

Save/Export Select UTF-8 Encoding to save or export using Unicode encoding, or select Default Encoding to save or export using the encoding form of the language currently used by your system.

Turn the export encoding warning off or on

- 1 Select Edit > Preferences (Windows) or Flash > Preferences (Macintosh), and select Warnings in the Category list.
- 2 Select or deselect Warn On Encoding Conflicts When Exporting ActionScript Files.

Encoding for imported and exported scripts

You can set ActionScript preferences to specify the type of encoding to use in imported or exported ActionScript files. UTF-8 Encoding is 8-bit Unicode format, which lets you include text in multiple languages in your file; Default Encoding is the encoding supported by the language your system is currently using, also called the *traditional code page*.

Important: When you use a non-English application on an English system, the Test Movie command fails if any part of the SWF file path has characters that cannot be represented by using the Multibyte Character Sets (MBCS) encoding scheme. For example, Japanese paths, which work on a Japanese system, won't work on an English system. Be sure to use English-only path names on English systems. All areas of the application that use the Test Movie player are subject to this limitation.


Pin scripts in the Actions panel

If you don't organize the code within your FLA file into one central location, or if you're using behaviors, you can pin individual scripts in the Actions panel to move among them more easily. To *pin* a script means that you can keep the location of the code open in the Actions panel, and easily click between open scripts. This can be especially useful when debugging.

In the following figure, the script associated with the current location on the timeline is on Frame 1 of the layer named Cleanup. (The tab at the far left always follows your location along the timeline.) That script is also pinned (it is shown as the rightmost tab). Two other scripts are pinned: one on Frame 1 and the other on Frame 15 of the layer named Intro. You can move among the pinned scripts by clicking the tabs or by using keyboard shortcuts. Moving among pinned scripts does not change your current position on the timeline.



A pinned script


 *If the content in the Script pane doesn't change to reflect the location you select on the timeline, the Script pane is probably showing a pinned script. Click the tab at the lower left of the Script pane to show the script associated with your location along the timeline.*

See also


“[Debugging ActionScript 3.0](#)” on page 360

“[Debugging ActionScript 1.0 and 2.0](#)” on page 350

Pin a script

- 1 Click the Timeline so the script appears in a tab at the lower left of the Script pane in the Actions panel.
- 2 Do one of the following:
 - Click the Pushpin icon to the right of the tab.
 - Right-click (Windows) or Control-click (Macintosh) the tab, and select Pin Script.
 - From the panel menu  (at the upper-right corner of the Actions panel), select Pin Script.

Unpin a script

- If a pinned script appears in a tab at the lower left of the Script pane in the Actions panel, click the Pushpin icon on the right of the tab.
- Right-click (Windows) or Control-click (Macintosh) a tab, and select Close Script or Close All Scripts.
- From the panel menu  (at the upper-right corner of the Actions panel), select Close Script or Close All Scripts.


Keyboard shortcuts for pinned scripts

When the insertion point is in the Script pane, use the following keyboard shortcuts to work with pinned scripts.

Action	Windows shortcut key	Macintosh shortcut key
Pin script	Control+= (equal sign)	Command+=
Unpin script	Control+- (minus sign)	Command+-
Move focus to tab on the right	Control+Shift+. (period)	Command+Shift+.
Move focus to tab on the left	Control+Shift+, (comma)	Command+Shift+,
Unpin all scripts	Control+Shift+- (minus sign)	Command+Shift+-

Insert target paths

Many script actions are meant to affect movie clips, buttons, and other symbol instances. In your code, you can reference symbol instances on a timeline by inserting a *target path*—the address of the instance you want to target. You can set either an absolute or relative target path. An absolute path contains the entire address of the instance. A relative path contains only the part of the address that is different from the address of the script itself in the FLA file, and will no longer work if the script is moved to another location.

- 1 In the Actions panel, click an action in your script.
- 2 Click Target Path .
- 3 Enter the path to the target instance, or select the target from the list.
- 4 Select either the Absolute or Relative path option.


Use code hints

When you work in the Actions panel or Script window, the software can detect what action you are entering and display a *code hint*. There are two types of code hint: a tooltip that contains the complete syntax for that action, and a pop-up menu that lists possible ActionScript elements, such as method or property names (sometimes referred to as a form of *code completion*).

Code hints are enabled by default. By setting preferences, you can disable code hints or determine how quickly they appear. When code hints are disabled in preferences, you can still manually display a code hint for a specific command.

Note: *If you cannot display code hints for a variable or object you created in ActionScript 2.0, but code hints are enabled in ActionScript preferences, make sure you named your variable or object with the proper suffix or that you used strict typing for your variable or object.*

Specify preference settings for automatic code hints

- In the Actions panel or Script window, select Edit > Preferences (Windows) or Flash > Preferences (Macintosh), click ActionScript in the Category list, and then enable or disable Code Hints.
- Select Preferences from the panel menu  (at the upper right of the Actions panel), and enable or disable Code Hints in the ActionScript preferences.

Specify a delay for code hints

- 1 In the Actions panel or Script window, select Edit > Preferences (Windows) or Flash > Preferences (Macintosh).
- 2 Click ActionScript in the Category list.
- 3 Use the slider to select an amount of delay, in seconds.

Use tooltip-style code hints

- 1 Display the code hint by typing an opening parenthesis (“(“ after an element that requires parentheses (for example, after a method name, a command such as `if` or `do..while`, and so on).

```
if (
  1 of 2 if ( condition ) {
}

my_array.splice(
  Array.splice( index, count, element1, ..., elementN )
```

Opening parentheses invoke the code hints

- 2 Enter a value for the parameter.

For multiple parameters, separate the values with commas. For functions or statements, such as the `for` loop, separate the parameters with semicolons.

Overloaded commands (functions or methods that can be invoked with different sets of parameters) such as `gotoAndPlay()` or `for` display an indicator that lets you select the parameter you want to set. To select the parameter, click the small arrows or press Control+Left Arrow and Control+Right Arrow.

```
for (
  1 of 2 for ( init; condition; next ) {
}

for (
  2 of 2 for ( $iterator$ in $object$ ) {
}
```

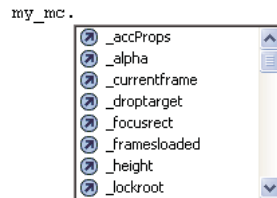
A code hint with multiple sets of parameters

3 To dismiss the code hint, do one of the following:

- Type a closing parenthesis “)”.
- Click outside the statement.
- Press Escape.

Use menu-style code hints:

1 Display the code hint by typing a period after a variable or object name.



Menu-style code hints

2 To navigate through the code hints, use the Up and Down Arrow keys.

3 To select an item in the menu, press Enter or Tab, or double-click the item.

4 To dismiss the code hint, do one of the following:

- Select one of the menu items.
- Click above or below the menu window.
- Type a closing parenthesis “)” if you’ve already typed an opening parenthesis “(“.
- Press Escape.

Manually display a code hint

1 Click in a code location where code hints can appear, such as in the following locations:

- After the dot (.) following a statement or command, where a property or method must be entered
- Between parentheses [()] in a method name

2 Do one of the following:

- Click Show Code Hint (🔍) in the Actions panel or Script window toolbar.
- Press Control+Spacebar (Windows) or Command+Spacebar (Macintosh).
- From the panel menu ☰ (at the upper-right corner of the Actions panel), select Show Code Hint.

Reload code hints without restarting the software

❖ From the panel menu ☰ (at the upper-right corner of the Actions panel), select Reload Code Hints.

You might need to do this if you customize Script Assist mode by writing custom methods.

Triggering code hints

You can trigger code hints in various ways.

Strict typing for objects

When you use ActionScript 2.0 and use strict typing for a variable that is based on a built-in class (such as Button, Array, and so on), the Script pane displays code hints for the variable. For example, suppose you enter the following two lines of code:

```
var foo:Array = new Array();
foo.
```

As soon as you enter the period (.), Flash displays a list of methods and properties available for Array objects in a pop-up menu, because you typed the variable as an array.


Suffixes and code hints

If you use ActionScript 1.0 or you want to display code hints for objects without strictly typing them, add a suffix to the name of each object when you create it. For example, the suffix that triggers code hinting for the Camera class is `_cam`. Suppose you type the following code:

```
var my_array = new Array();
var my_cam = Camera.get();
```

If you then enter `my_cam` followed by a period, code hints for the Camera object appear.

For objects that appear on the Stage, use the suffix in the Instance Name box in the Property inspector. For example, to display code hints for MovieClip objects, use the Property inspector to assign instance names with the `_mc` suffix to all MovieClip objects. Then, whenever you type the instance name followed by a period, code hints appear.

 *Although suffixes are not required for triggering code hints when you use strict typing for an object, using suffixes consistently helps make your code understandable.*

The following table lists the suffixes that trigger code hints in ActionScript 2.0:

Object type	Variable suffix
Array	<code>_array</code>
Button	<code>_btn</code>
Camera	<code>_cam</code>
Color	<code>_color</code>
ContextMenu	<code>_cm</code>
ContextMenuItem	<code>_cmi</code>
Date	<code>_date</code>
Error	<code>_err</code>
LoadVars	<code>_lv</code>
LocalConnection	<code>_lc</code>
Microphone	<code>_mic</code>
MovieClip	<code>_mc</code>
MovieClipLoader	<code>_mcl</code>
PrintJob	<code>_pj</code>
NetConnection	<code>_nc</code>
NetStream	<code>_ns</code>

Object type	Variable suffix
SharedObject	_so
Sound	_sound
String	_str
TextField	_txt
TextFormat	_fmt
Video	_video
XML	_xml
XMLNode	_xmlnode
XMLSocket	_xmlsocket

Comments and code hints

You can also use ActionScript comments to specify an object's class for code hints. In the following example, a comment tells ActionScript that the class of the `theObject` instance is `Object`, and so on.

```
// Object theObject;// Array theArray;// MovieClip theMC;
```

If you subsequently enter `theMC` followed by a period, code hints that display the list of `MovieClip` methods and properties appear. If you enter `theArray` followed by a period, code hints that display the list of `Array` methods and properties appear, and so on.

Instead of this technique, however, Adobe recommends that you use strict data typing or suffixes, because these techniques enable code hints automatically and make your code more understandable.

See also

Data types

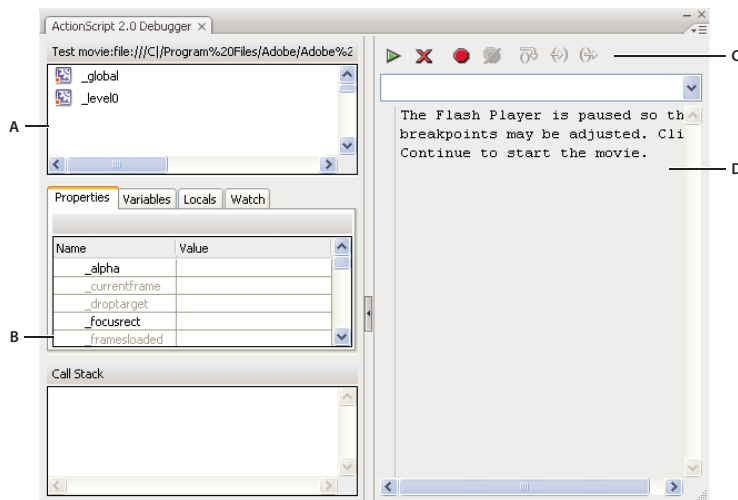
Data type descriptions

Debugging ActionScript 1.0 and 2.0

Debugging your ActionScript 1.0 and 2.0 scripts

The ActionScript 2.0 debugger helps you find ActionScript 1.0 and 2.0 errors while your SWF file runs in Flash Player. When using Flash to debug ActionScript 1.0 and 2.0, you view your SWF files in the debugger version of Flash Player, which is installed automatically with Flash. To install a stand-alone debugger version of Flash Player, run the installer in the */Flash installed directory/Players/Debug/* directory.

The ActionScript 2.0 Debugger shows a hierarchical display list of movie clips currently loaded in Flash Player. Using the ActionScript 2.0 Debugger, you can display and modify variable and property values as the SWF file plays, and you can use breakpoints to stop the SWF file and step through ActionScript code line by line. You can then go back to your scripts and edit them so that they produce the correct results.



A. Display list B. Properties list C. Tool bar D. Code view

You can resize the regions of the Debugger panel. When your pointer changes between each region, drag to resize the Display list, Watch list, and code view. You can also click the vertical bar to expand either side of the Debugger to full size.

After it's activated, the Debugger status bar displays the URL or local path of the file, tells whether the file is running in the test environment or from a remote location, and shows a live view of the movie clip display list. When movie clips are added to or removed from the file, the display list reflects the changes immediately.

Note: *The ActionScript 2.0 and 3.0 debuggers have significant differences. For more information about the ActionScript 3.0 debugger, see “Debugging ActionScript 3.0” on page 360.*

The Test Movie command and keyboard controls

When you use the Control > Test Movie command to test SWF files that implement keyboard controls (tabbing, keyboard shortcuts created using `Key.addListener()`, and so on), select Control > Disable Keyboard Shortcuts in the Test Movie window. Selecting this option prevents the authoring environment from “grabbing” keystrokes, and lets them pass through to the player. For example, in the authoring environment, Control+U opens the Preferences dialog box. If your script assigns Control+U to an action that underlines text onscreen, when you use Test Movie, pressing Control+U opens the Preferences dialog box instead of running the action that underlines text. To let the Control+U command pass through to the player, you must select Control > Disable Keyboard Shortcuts in the Test Movie window.

Important: *When you use a non-English application on an English system, the Test Movie command fails if any part of the SWF file path has characters that cannot be represented with the MBCS encoding scheme. For example, Japanese paths on an English system do not work.*

Other debugging tools

Flash also provides the following debugging tools:

- The Compiler Errors panel, which shows errors encountered when Flash compiles your scripts
- The Output panel, which shows runtime error messages, and lists of variables and objects
- The `trace()` statement, which sends programming notes and values of expressions to the Output panel
- The `throw` and `try..catch..finally` statements, which let you test and respond to runtime errors from within your script

Debug a local SWF file

- 1 Open the FLA document.
- 2 Select Debug > Debug Movie.

This command exports the SWF file with debugging information (the SWD file). It opens the Debugger and opens the SWF file in the test environment. The SWD file is used to debug ActionScript, and contains information that lets you use breakpoints and step through code.

Debug a remote ActionScript 2.0 SWF file

You can debug a remote SWF file by using the stand-alone, ActiveX, or plug-in version of the Debug Flash Player, which you can find in the *Flash install directory/Players/Debug/* directory.

To permit remote debugging of the file, enable debugging in the Publish settings. You can also publish your file with a debugging password to ensure that only trusted users can debug it.

As in JavaScript or HTML, users can view client-side variables in ActionScript. To store variables securely, send them to a server-side application instead of storing them in your file. However, as a developer, you may have other trade secrets, such as movie clip structures, that you do not want to reveal. You can use a debugging password to protect your work.

Enable remote debugging of a SWF file and set a debugging password

- 1 Open the FLA file.
- 2 Select File > Publish Settings.
- 3 On the Flash tab of the Publish Settings dialog box, select Permit Debugging.
- 4 To set a password, enter a password in the Password box.

After you set this password, no one can download information to the Debugger without the password.

- 5 Close the Publish Settings dialog box, and select one of the following commands:

- Debug > Debug Movie
- File > Export > Export Movie
- File > Publish

Flash creates a debugging file, with the extension .swd, and saves it in the same directory as the SWF file. The SWD file is used to debug ActionScript, and contains information that lets you use breakpoints and step through code.

- 6 Upload the SWF file and the SWD file to the same directory on your web server, or leave it on the local machine to perform a remote debug session on the localhost.

If the SWD file is not in the same directory as the SWF file, you can still debug remotely; however, the Debugger has no breakpoint information, so you can't step through code.

- 7 In Flash, select Debug > Begin Remote Debug Session > ActionScript 2.0.

Flash opens the ActionScript 2.0 Debugger panel and waits for a debug Flash Player to connect. You have 2 minutes to start the debug Flash Player. If more than 2 minutes elapse, repeat this step.

- 8 Open the SWF file in the debug version of the Flash Player plugin, ActiveX control, or stand-alone player. The debug stand-alone player is located in the *Flash install directory/Players/Debug/* directory.

The debug session begins when the debug player connects to the Flash ActionScript 2.0 Debugger panel.

Activate the Debugger from a remote location

- 1 Open the Flash authoring application if it is not already open.
- 2 Select Debug > Begin Remote Debug Session > ActionScript 2.0.
- 3 In a browser or in the debugger version of the stand-alone player, open the published SWF file from the remote location. Be sure the SWD file is in the same folder as the SWF file.

If the Remote Debug dialog box does not appear, right-click (Windows) or Control-click (Macintosh) in the SWF file to display the context menu, and select Debugger.

- 4 In the Remote Debug dialog box, select Localhost or Other Machine:
 - Select Localhost if the debugger version of Flash Player and the Flash authoring application are on the same computer.
 - Select Other Machine if the debugger version of Flash Player and the Flash authoring application are not on the same computer. Enter the IP address of the computer running the Flash authoring application.
 - Select Don't show this dialog at launch to prevent the Remote Debug dialog box from prompting the user for a debugger location if one cannot be found. This option is selected by default.
- 5 Enter your debugging password if you set one.

The display list of the SWF file appears in the Debugger. If the SWF file doesn't play, the Debugger might be paused, so click Continue to start it.

Display and modify the values of variables in the Debugger

The Variables tab in the Debugger shows the names and values of any global and timeline variables that are selected in the SWF file's display list. If you change the value of a variable on the Variables tab, the change is reflected in the SWF file while it runs. For example, to test collision detection in a game, you can enter the variable value to position a ball in the correct location next to a wall.

The Locals tab in the Debugger shows the names and values of any local variables that are available in the line of ActionScript where the SWF file is currently stopped, at a breakpoint or anywhere else within a user-defined function.

See also

[“List a SWF file's objects and variables”](#) on page 355

Display a variable and its value

- 1 In the Debugger's display list, select the movie clip containing the variable. (To display global variables, select the `_global` clip in the display list.)
- 2 Click the Variables tab.

The display list updates automatically as the SWF file plays.

Note: If a movie clip is removed from the SWF file at a specific frame, that movie clip, along with its variable and variable name, is also removed from the display list in the Debugger. However, if its variable is marked for the Watch list, you can still view it in the Watch tab.

Modify a variable value

- ❖ On the Debugger panel Variables tab, double-click the value, and enter a new value.

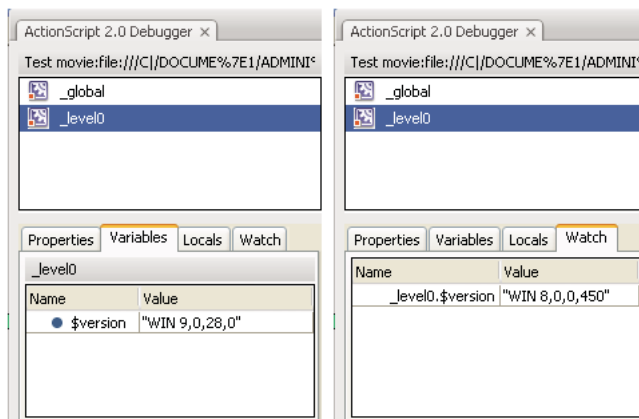
Enter a string (any value surrounded by quotation marks), a number, or a Boolean value (`true` or `false`). You cannot enter an expression (for example, `eval ("name: " + i) or x + 2`).

Note: To write the value of an expression to the Output panel in the test environment, use the `trace()` statement.

Use the Watch list

To monitor a set of critical variables, you can mark them for the Watch list. The Watch list shows the absolute path to the variable and the value. You can also enter a new variable value in the Watch list. The Watch list shows only variables that you access by using an absolute target path, such as `_global` or `_root`.

If you add a local variable to the Watch list, its value appears only when Flash Player stops at a line of ActionScript that is within the variable's scope. All other variables appear while the SWF file is playing. If the Debugger can't find the value of the variable, the value is listed as undefined.



Variables marked for the Watch list and variables in the Watch list

Add variables to the Watch list

- On the Variables or Locals tab, right-click (Windows) or Control-click (Macintosh) a selected variable; then select Watch from the context menu. A blue dot appears next to the variable.
- On the Watch tab, right-click (Windows) or Control-click (Macintosh) and select Add from the context menu. Double-click in the name column, and enter the target path to the variable name.

Remove variables from the Watch list

- ❖ On the Watch tab or the Variables tab, right-click (Windows) or Control-click (Macintosh) and select Remove from the context menu.

Display movie clip properties and change editable properties

The Debugger's Properties tab shows all the property values of any movie clip on the Stage. If you change a value, you can see its effect in the SWF file while it runs. (Some movie clip properties are read-only and cannot be changed.)

Display a movie clip's properties in the Debugger

- 1 Select a movie clip from the display list.
- 2 Click the Properties tab in the Debugger.

Modify a property value

- ❖ In the Debugger's Properties tab, double-click the value, and enter a new value.

Enter a string (any value surrounded by quotation marks), a number, or a Boolean value (`true` or `false`). You cannot enter an expression (for example, `x + 50`), or object or array values (for example, `{id: "rogue"}` or `[1, 2, 3]`).

Note: To write the value of an expression to the Output panel in the test environment, use the `trace()` statement.

List a SWF file's objects and variables

To list a SWF file's objects, use the List Objects command, which is useful for finding the correct target path and instance names. To list a SWF file's variables, use the List Variables command, with which you can find a variable's name and target path.

Selecting the List Objects or List Variables command clears the contents of the Output panel. If you do not want to lose this information, select Save to File from the Output Panel menu before selecting the command.

List a SWF file's objects

In the test environment, the List Objects command shows the level, frame, object type (shape, movie clip, or button), target paths, and instance names of movie clips, buttons, and text fields in a hierarchical list in the Output panel. (It does not show all ActionScript data objects.)

- 1 If your SWF file is not running in the test environment, select Control > Test Movie.
- 2 Select Debug > List Objects.

A list of all the objects currently on the Stage appears in the Output panel. The list does not update automatically as the SWF file plays; you must select the List Objects command each time you want to send the information to the Output panel.

List a SWF file's variables in the Output panel

In the test environment, the List Variables command shows a list of all the variables in the SWF file. Global variables (those declared with the `_global` identifier) appear at the top of the List Variables output in a Global Variables section, and each variable has a `_global` prefix.

In addition, the List Variables command shows getter/setter properties—properties that are created with the `Object.addProperty()` method and start `get` or `set` methods. In the Output panel, the value of a getter/setter property is prefixed with `[getter/setter]`. The software determines the value that appears for a getter/setter property by evaluating the `get` function.

The list does not update automatically as the SWF file plays; you must select the List Variables command each time you want to send the information to the Output panel.

- 1 Create a new FLA document. For example, you might call it `listvariables fla`.
- 2 Add the following ActionScript 2.0 in the Actions panel:

```
_global.myName = "Buster";  
var myCatSays:String = "meow";  
var myNum:Number = 313;  
var myArray:Array = ["one", "two", "three"];
```

- 3 Select Control > Test Movie.
- 4 Select Debug > List Variables in the test environment.

A list of all the variables currently in the SWF file appears in the Output panel. The following example shows the variables that would be output from the code listed above in step 2:

```
Global Variables:
  Variable _global.myName = "Buster"
Level #0:
Variable _level0.$version = "WIN 9,0,29,3"
Variable _level0.myCatSays = "meow"
Variable _level0.myNum = 313
Variable _level0.myArray = [Object #1, class 'Array'] [0:"one",
  1:"two",
  2:"three"
]
```

Set and remove breakpoints

A breakpoint lets you stop an application at a specific line of ActionScript. You can use breakpoints to test possible trouble spots in your code. For example, if you've written a set of `if...else if` statements and can't determine which one is executing, you can add a breakpoint before the statements and examine them one by one (step through them) in the Debugger.

You can set breakpoints in the Actions panel, Script window, or Debugger. Breakpoints set in the Actions panel are saved with the FLA file. Breakpoints set in the Debugger and Script window are not saved in the FLA file and are valid only for the current debugging session.


Important: *If you set breakpoints in the Actions panel or Script window and click Auto Format, check your breakpoints. If the Auto Format command removed empty lines, your ActionScript might be moved to a different line. It's a good idea to autoforamt your scripts before you set breakpoints.*

You can view breakpoints in both the Debugger and the Script window by setting them in either one of those windows. For this to work, the path to the AS file must be the same in both windows.

Do not set breakpoints on comments or empty lines; these breakpoints are ignored.

Set or remove a breakpoint in the Actions panel or Script window

During a debugging session, do one of the following:

- Click in the left margin of the Script pane. A red dot indicates a breakpoint.
- Click Debug Options  button.
- Right-click (Windows) or Control-click (Macintosh) to display the context menu, and select Set Breakpoint, Remove Breakpoint, or Remove Breakpoints In This File. (In the Script window, you can also select Remove Breakpoints In All AS Files.)
- Press Control+Shift+B (Windows) or Command+Shift+B (Macintosh).

Note: *In some previous versions of Flash, clicking in the left margin of the Script pane selected the line of code; now it adds or removes a breakpoint. To select a line of code, use Control-click (Windows) or Command-click (Macintosh).*

Set and remove breakpoints in the Debugger

- Click in the left margin of the code pane. A red dot indicates a breakpoint.
- Click Toggle Breakpoint or Remove All Breakpoints above the code view.
- Right-click (Windows) or Control-click (Macintosh) to display the context menu, and select Set Breakpoint, Remove Breakpoint, or Remove All Breakpoints in the File.

- Press Control+Shift+B (Windows) or Command+Shift+B (Macintosh).

The breakpoints XML file

When you work with breakpoints in the Script window, the `AsBreakpoints.xml` file lets you store breakpoint information. This file is written to the Local Settings directory, in the following locations:

Windows *Hard Disk\Documents and Settings\User\Local Settings\Application Data\Adobe\Flash CS3\language\Configuration\Debugger*

Macintosh *Macintosh HD/Users/User/Library/Application Support/Adobe Flash CS3/Configuration/Debugger/*

Here is an example of an `AsBreakpoints.xml` file:

```
<?xml version="1.0"?>
<flash_breakpoints version="1.0">
  <file name="c:\tmp\myscript.as">
    <breakpoint line="10"></breakpoint>
    <breakpoint line="8"></breakpoint>
    <breakpoint line="6"></breakpoint>
  </file>
  <file name="c:\tmp\myotherscript.as">
    <breakpoint line="11"></breakpoint>
    <breakpoint line="7"></breakpoint>
    <breakpoint line="4"></breakpoint>
  </file>
</flash_breakpoints>
```

The XML file consists of the following tags:

flash_breakpoints This node has a `version` attribute, which specifies the version of the XML file. Flash 8 is version 1.0.

file A child node of `flash_breakpoints`. This node has a `name` attribute, which specifies the name of the file that contains breakpoints.

breakpoint A child node of `file`. This node has a `line` attribute, which specifies the line number that contains the breakpoint.

The `AsBreakpoints.xml` file is read when you start Flash, and regenerated when you quit. `AsBreakpoints.xml` is used to keep track of the breakpoints between development sessions.

Stepping through lines of code


After you set breakpoints in a script and click Continue in the Debugger, you can step through lines of code—that is, control how the Debugger moves through statements and functions.

For example, in the following ActionScript 2.0 code, suppose a breakpoint is set inside a button on the `myFunction()` line:

```
on(press) {
    myFunction();
}
```

When you click the button, Flash Player reaches the breakpoint and pauses. You can now bring the Debugger to the first line of `myFunction()` wherever it is defined in the document. You can also continue through or exit out of the function.


As you step through lines of code, the values of variables and properties change in the Variables, Locals, Properties, and Watch tabs. A yellow arrow on the left side of the Debugger's code view indicates the line at which the Debugger stopped. Use the following buttons along the top of the code view:


Step In  Advances the Debugger into a function. (If a line does not contain a user-defined function, Step In advances to the next line.)

In the following example, if you place a breakpoint at line 7 and click Step In, the Debugger advances to line 2, and another click of Step In advances you to line 3.


```
1 function myFunction() {
2   x = 0;
3   y = 0;
4 }
5
6 mover = 1;
7 myFunction();
8 mover = 0;
```

Note: The numbers in this code snippet denote line numbers. They are not part of the code.

Step Out  Advances the Debugger out of a function. This button works only if you are currently stopped in a user-defined function; it moves the yellow arrow to the line that follows the function call. In the previous example, if you place a breakpoint at line 3 and click Step Out, the Debugger moves to line 8. Clicking Step Out at a line that is not within a user-defined function is the same as clicking Continue. For example, if you stop at line 6 and click Step Out, the player continues to execute the script until it encounters a breakpoint.

Step Over  Advances the Debugger over a line of code. This button moves the yellow arrow to the next line in the script. In the previous example, if you are stopped at line 7 and click Step Over, you advance directly to line 8 without stepping through `myFunction()`, although the `myFunction()` code still executes.

Continue  Leaves the line at which the player is stopped and continues playing until a breakpoint is reached.

End Debug Session  Makes the Debugger inactive but continues to play the SWF file in Flash Player.

Control compiler warnings

You can control the types of compiler warnings that the ActionScript compiler generates in the Compiler Errors panel. When the compiler reports an error, you can double click on the error to navigate to the line of code that caused the error.

- 1 Select File > Publish Settings.
- 2 Click the Flash tab.
- 3 Click the ActionScript Settings button.
- 4 Select among the Errors options:
 - Strict Mode causes warnings to be reported as errors, which means that compilation will not succeed if those errors exist.
 - Warnings Mode causes extra warnings to be reported that are useful for discovering incompatibilities when updating ActionScript 2.0 code to ActionScript 3.0.

Output panel overview


When you test a SWF file, the Output panel can show information to help you troubleshoot your SWF file. To show this information, add `trace()` statements to your code or use the List Objects and List Variables commands.

If you use the `trace()` statement in your scripts, you can send specific information to the Output panel as the SWF file runs. This could include notes about the SWF file's status or the value of an expression. For more information, see the `trace()` function in the *ActionScript 2.0 Language Reference* at www.adobe.com/go/learn_fl_cs4_as2lr_en.

Display or hide the Output panel

❖ Select Window > Output or press F2.

Work with the contents of the Output panel

❖ In the Output Panel menu , select a command. Here is a partial list:

Copy Copies all the contents of the Output panel to the computer's Clipboard. To copy a selected portion of the output, select the area you want to copy and then select Copy.

Save To File Saves the Output panel contents to a text file.

Filter Level Select None to prevent any information from appearing in the Output panel; select Verbose to send all information to the Output panel.

Use the trace statement

A `trace()` statement sends specific information to the Output panel. For example, while testing a SWF file, you can send results to the Output panel when a button is pressed or a frame plays. The `trace()` statement is similar to the JavaScript `alert` statement.

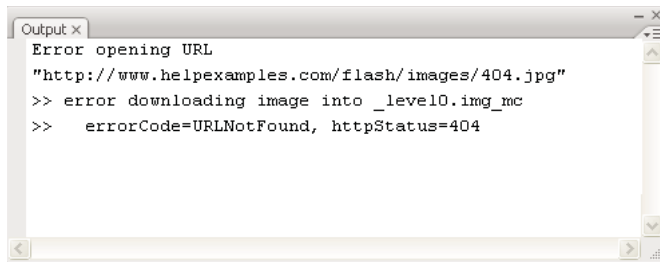
When you use the `trace()` statement in a script, you can use expressions as parameters. The value of an expression appears in the Output panel in the test environment.

1 Select a frame in the Timeline and add a `trace()` statement. For example, you might select Frame 1 and add the following ActionScript 2.0 code:

```
this.createEmptyMovieClip("img_mc", 10);
var mclListener:Object = new Object();
mclListener.onLoadInit = function(target_mc:MovieClip) {
    trace(target_mc+" loaded in "+getTimer()+" ms");
};
mclListener.onLoadError = function(target_mc:MovieClip, errorCode:String,
httpStatus:Number) {
    trace(">> error downloading image into "+target_mc);
    trace(">>\t errorCode="+errorCode+", httpStatus="+httpStatus);
};
var img_mcl:MovieClipLoader = new MovieClipLoader();
img_mcl.addListener(mclListener);
img_mcl.loadClip("http://www.helpexamples.com/flash/images/404.jpg", img_mc);
```

2 Select Control > Test Movie to test the SWF file.

The Output panel displays the results of the `trace()` statement. For example, the panel might display the following message:

A screenshot of the 'Output' panel in an IDE. The panel title is 'Output x'. The text inside the panel reads: 'Error opening URL', followed by a quoted URL: '"http://www.helpexamples.com/flash/images/404.jpg"', then '>> error downloading image into _level0.img_mc', and finally '>> errorCode=URLNotFound, httpStatus=404'. The panel has a scroll bar on the right and a close button in the top right corner.

```
Output x
Error opening URL
"http://www.helpexamples.com/flash/images/404.jpg"
>> error downloading image into _level0.img_mc
>>  errorCode=URLNotFound, httpStatus=404
```

Output panel

Navigate to errors in code

When Flash encounters an error in ActionScript code, either during compiling or execution, it reports the error in the Compiler Errors panel. From the Compiler Errors panel, you can navigate to the line of code that caused the error.

- ❖ Double click the error in the Compiler Errors panel.

Debugging text field objects

To obtain debugging information about TextField objects, you can use the Debug > List Variables command or the Debug > List Objects command in the test environment. When you use Debug > List Variables, the Output panel uses the following conventions to show TextField properties:

- No more than four properties appear on a line.
- A property with a string value appears on a separate line.
- Color properties appear as hexadecimal numbers (0x00FF00).
- The properties appear in the following order: variable, text, htmlText, html, textWidth, textHeight, maxChars, borderColor, backgroundColor, textColor, border, background, wordWrap, password, multiline, selectable, scroll, hscroll, maxscroll, maxhscroll, bottomScroll, type, embedFonts, restrict, length, tabIndex, autoSize.

The Debug > List Objects command lists TextField objects. If an instance name is specified for a text field, the Output panel shows the full target path including the instance name in the following form:

```
Target = "target path"
```

See also

[“Control compiler warnings”](#) on page 358

Debugging ActionScript 3.0

About the ActionScript 3.0 debugger

Flash includes a separate debugger for ActionScript 3.0 that operates somewhat differently from the ActionScript 2.0 debugger. The ActionScript 3.0 debugger only works with ActionScript 3.0 FLA and AS files. FLA files must have publish settings set to Flash Player 9. When you initiate an ActionScript 3.0 debugging session, Flash launches the stand-alone debug version of Flash Player to play the SWF file. The debug Flash player plays the SWF in a separate window from the Flash authoring application window.

The ActionScript 3.0 debugger converts the Flash workspace to a debug workspace that displays panels that are used for debugging, including the Actions panel and/or Script window, the Debug console, and the Variables panel. The Debug console displays the call stack and contains tools for stepping through scripts. The Variables panel displays the variables in the current scope with their values and allows you to update those values yourself.

Enter debugging mode

The way you begin a debugging session depends on the type of file you are working on. During a debugging session, Flash interrupts the execution of ActionScript when it encounters a breakpoint or a runtime error.

When Flash initiates a debug session, it adds special information to the SWF file that it exports for the session. This information allows the debugger to provide the specific line numbers in the code where errors are encountered.

You can include this special debugging information in all SWF files created from a specific FLA file in the Publish settings. This allows you to debug the SWF file even if you do not explicitly initiate a debug session. This debugging information makes the SWF file slightly larger.

Start debugging from a FLA file

- ❖ Select Debug > Debug Movie.

Start debugging from an ActionScript 3.0 AS file

- 1 With the ActionScript file open in the Script window, select the FLA file that the ActionScript file should be compiled with from the Target menu at the top of the Script window. The FLA file must also be open in Flash to appear in this menu.
- 2 Select Debug > Debug Movie.

Add debugging information to all SWF files created from a FLA file

- 1 With the FLA file open, select File > Publish Settings.
- 2 In the Publish Settings dialog box, click the Flash tab.
- 3 Select Permit Debugging.

Exit debugging mode

- ❖ Click the End Debug Session button in the Debug Console.

Set and remove breakpoints

Add breakpoints to ActionScript code to interrupt the execution of the code. After execution is interrupted, you can step through and execute the code line by line, view different sections of your ActionScript, view the values of variables and expressions, and edit variable values.

Note: Breakpoints cannot be added to ASC (ActionScript for Communication) or JSFL (Flash JavaScript) files.

Set a breakpoint

- ❖ In the Actions panel or Script window, click in the left margin next to the line of code where you want the breakpoint to appear.

Remove a breakpoint

- ❖ In the Actions panel or Script window, click on the breakpoint to remove.

Step through lines of code

After the ActionScript execution is interrupted at a breakpoint or runtime error, you can step through the code line by line, choosing to step into function calls or step over them. You can also choose to continue executing the code without stepping.

Step into code line by line

- ❖ Click the Step In button in the Debug Console.

Step over a function call

- ❖ Click the Step Over button in the Debug Console.

Step out of a function call

- ❖ Click the Step Out button in the Debug Console.

Resume normal code execution

- ❖ Click the Continue button in the Debug Console.

Display and examine scripts in the call stack

When code execution stops in the debugger, you can view the call stack in the Debug Console and display the scripts containing the functions in the call stack. The call stack shows the current list of nested function calls that are waiting to complete execution.

You can view the individual scripts that contain each function.

- ❖ In the Debug Console panel, double click the name of the script in the call stack.

Display and modify variable values

View and edit the values of variables and properties in the Variables panel.

View a variable value

- 1 In the Variables panel, select the types of variables to display from the Panel menu.
 - Show Constants displays the values constants (variables having a fixed value).
 - Show Statics displays variables that belong to the class, rather than to instances of the class.
 - Show Inaccessible Member Variables displays variables that are not accessible to other classes or namespaces. This includes variables that are protected, private or internal to the namespace.
 - Show Additional Hexadecimal Display adds hexadecimal values wherever decimal values are displayed. This is mainly useful for color values. Hexadecimal values are not displayed for decimal values from 0 through 9.
 - Show Qualified Names displays variables types with both the package name and the class name.
- 2 Expand the tree view of the object structure of the FLA until you see the variable to view.

Edit the value of a variable

- 1 In the Variables panel, double click on the value of the variable.
- 2 Enter the new value for the variable and press Enter. The new value is used during subsequent code execution.

Control compiler warnings

Control the types of compiler warnings that the ActionScript compiler generates in the Compiler Errors panel. When the compiler reports an error, double click the error to navigate to the line of code that caused the error.

- 1 Select File > Publish Settings.
- 2 Click Flash.
- 3 Click the ActionScript Settings button.
- 4 Select among the Errors options:
 - Strict Mode reports warnings as errors, which means that compilation will not succeed if those errors exist.
 - Warnings Mode reports extra warnings that are useful for discovering incompatibilities when updating ActionScript 2.0 code to ActionScript 3.0.

Navigate to errors in code

When Flash encounters an error in ActionScript code, either during compiling or execution, it reports the error in the Compiler Errors panel. Navigate to the line of code that caused the error from the Compiler Errors panel.

- ❖ Double click the error in the Compiler Errors panel.

Debug a remote ActionScript 3.0 SWF file

With ActionScript 3.0, you can debug a remote SWF file by using the stand-alone, ActiveX, or plug-in version of the Debug Flash Player, which you can find in the *Flash install directory/Players/Debug/* directory. However, in the ActionScript 3.0 Debugger, remote debugging is limited to files located on the same localhost as the Flash authoring application, being played in the stand alone debug player, ActiveX control, or plugin.

To permit remote debugging of the file, enable debugging in the Publish settings. You can also publish your file with a debugging password to ensure that only trusted users can debug it.

As in JavaScript or HTML, users can view client-side variables in ActionScript. To store variables securely, send them to a server-side application instead of storing them in your file. However, as a developer, you may have other trade secrets, such as movie clip structures, that you do not want to reveal. You can use a debugging password to protect your work.

Enable remote debugging of a SWF file and set a debugging password

In ActionScript 3.0 FLA files, code in frame scripts cannot be debugged. Only code in external AS files can be debugged with the ActionScript 3.0 Debugger.

- 1 Open the FLA file.
- 2 Select File > Publish Settings.
- 3 On the Flash tab of the Publish Settings dialog box, select Permit Debugging.
- 4 Close the Publish Settings dialog box, and select one of the following commands:
 - File > Export > Export Movie
 - File > Publish
- 5 Leave the SWF file on the local machine to perform a remote debug session on the localhost, or upload it to your web server.

The SWF file contains no breakpoint information, so if you upload the file to a remote server you will not be able to step through code. Use the localhost to perform this task.

- 6 In Flash, select Debug > Begin Remote Debug Session > ActionScript 3.0.

Flash opens the ActionScript 3.0 Debugger and waits for a debug Flash Player to connect. You have 2 minutes to start the debug Flash Player. If more than 2 minutes elapse, repeat this step.

- 7 Open the SWF file in the debug version of the Flash Player plugin, ActiveX control, or stand-alone player. The debug stand-alone player is located in the *Flash install directory*/Players/Debug/ directory. Do not connect to a file on another machine, as debugger will not be able to receive any breakpoint information.

The debug session begins when the debug player connects to the Flash ActionScript 3.0 Debugger panel.

Activate the Debugger from a remote location

- 1 Open the Flash authoring application if it is not already open.
- 2 Select Debug > Begin Remote Debug Session > ActionScript 3.0.
- 3 In a browser or in the debugger version of the stand-alone player, open the published SWF file from the remote location.

If the Remote Debug dialog box does not appear, right-click (Windows) or Control-click (Macintosh) in the SWF file to display the context menu, and select Debugger.

- 4 In the Remote Debug dialog box, select Localhost and select the file to open.

The display list of the SWF file appears in the Debugger. If the SWF file doesn't play, the Debugger might be paused, so click Continue to start it.

Working with Flash and Flex

Creating components for Flex

In Adobe® Flash® CS4 Professional, you can create content for use as components in Adobe® Flex® applications. This content can include both visual elements and Adobe® ActionScript® 3.0 code.

Creating components in Flash for use in Flex allows you to take advantage of the flexible graphic design capabilities of Flash while still utilizing the capabilities of Flex.

In order to create Flex components in Flash, you must install the Flex Component Kit for Flash. You install the component kit using Adobe Extension Manager. Some versions of the component kit may not support all features of Adobe® Flash® CS4 Professional, so be sure to download the latest version of the component kit from www.adobe.com/go/flex_ck_en.

For more information about using Flex and Flash together, refer to the Flex documentation on the Adobe web site at www.adobe.com/go/learn_flexresources_en.

To create a Flex component in Flash:

- 1 Be sure you have Adobe Extension Manager installed. To download Extension Manager, visit the Extension Manager download page at www.adobe.com/go/extension_manager_dl_en.

By default, Extension Manager is installed with the Adobe Creative Suite applications.

- 2 Download and install the Flex Component Kit, available at www.adobe.com/go/flex_ck_en. Be sure to quit Flash before installing the component kit. For information about installing extensions with Adobe Extension Manager, see www.adobe.com/go/learn_extension_manager_en.

- 3 Launch Flash. Two new commands appear in the Commands menu, Convert Symbol to Flex Component and Convert Symbol to Flex Container.
- 4 In Flash, create a movie clip symbol containing the artwork and ActionScript 3.0 code you want to include in the Flex component. The content must be contained in a movie clip symbol before conversion to a Flex component.
- 5 Before converting the movie clip to a Flex component, be sure that it meets the following requirements for compatibility with Flex:
 - The frame rate of the FLA file should be 24 fps and should match the frame rate of any Flex projects that will make use of the component.
 - The registration point should be located at the 0, 0 point in the movie clip.

Note: To ensure that all content in the movie clip has a registration point of 0, 0, click the Edit Multiple Frames button at the bottom of the Timeline, select all frames in the movie clip timeline, select all of your content in all the frames, and move it to 0, 0 in the Property inspector.
- 6 Select the movie clip in the Library panel and choose Commands > Convert Symbol to Flex Component.
Flash converts the movie clip to a Flex component, changes its icon to a Flex icon in the Library, and imports the FlexComponentBase class compiled clip to the Library. Flash embeds the FlexComponentBase into the Flex component SCW file created in the next step.

Note the progress messages displayed in the Output panel while Flash converts the movie clip.
- 7 Choose File > Publish to create a SWC file containing the compiled Flex component. Flash also creates a SWF file from the main FLA file, but you can ignore the SWF file if you choose. The published component SWC file is now ready for use in Flex.
- 8 To use the SWC file in Flex, do one of the following:
 - Copy the SWC file from Flash and paste it into the bin folder of your Flex project.
 - Add the SWC file to library path of your Flex project. For more information, see the Flex Builder documentation at www.adobe.com/go/learn_flexresources_en.

Using Flex metadata

If you are writing ActionScript 3.0 code to be used in Flex, you can place metadata in the code to embed external files in any published SWF that includes the ActionScript code. Usually, these `[Embed]` metadata declarations are used to embed image files, fonts, individual symbols, or other SWF files into the SWF.

Remember that metadata is “data about data.” You add metadata to ActionScript on the line immediately preceding the line of code that the metadata applies to. The compiler then takes the metadata into account when compiling the line of code that follows it.

For example, to embed an image called `button_up.png` that is stored in the directory one level above the ActionScript file, you would use the following ActionScript:

```
[Embed("../button_up.png")]  
  
private var buttonUpImage:Class;
```

The `[Embed]` metadata tag tells the compiler to embed the file named `button_up.png` in the SWF file and that the file should be associated with the variable named `buttonUpImage`.

For more information about embedding assets with metadata in Flex, see Embedding Assets in the Flex 3 Developer Guide at www.adobe.com/go/learn_flexresources_en.

If you use a feature that requires the Flex SDK, such as `[Embed]` metadata, at compile time Flash prompts you to add the Flex.SWC file to the Library path of your FLA file. The Flex.SWC file contains compiled classes needed to support Flex metadata. Click Update Library Path in the dialog box to add Flex.SWC to the Library path. You can also add the Flex.SWC file to the Library path in the ActionScript publish settings later.

Printing at runtime

Printing from SWF files at runtime using ActionScript 2.0

You can add printing functionality to Flash documents that lets users print from Flash Player. You can use the ActionScript® `PrintJob` class, or you can use the `print()` or `printAsBitmap()` ActionScript functions. Users can also access the Flash Player context menu and select the Print command there. Additionally, users can print from a browser, rather than from Flash Player, by selecting a command such as File > Print from the browser window. However, printing from Flash Player directly, rather than from a browser window Print menu, offers several advantages, including the following:

- Users can print all frames or certain frames that you've labeled as printable from Flash Player. Additionally, you can set the print area of a frame.
- You can specify that content print as vector graphics (to take advantage of higher resolution) or as bitmaps (to preserve transparency and color effects).
- The ActionScript `PrintJob` object improves upon the `print()` and `printAsBitmap()` functions by adding the ability to print dynamically rendered pages as a single print job. The `PrintJob` object also provides the user's printer settings, which can be used to format reports specifically for the user.
- Flash Player versions earlier than 4.0.25 (Windows) or 4.0.20 (Macintosh) do not support printing frames directly. Flash Player 7 and later supports the `PrintJob` class.

Note: For information on printing from SWF files at runtime using ActionScript 3.0, see *Printing in Programming ActionScript 3.0*.

Controlling runtime printing

To control what users can print, remember the following items as you set up documents and movie clips for printing:

- Adjust the page layout in any frames that you designate as printable to match the desired printed output. Using Flash Player, you can print all shapes, symbols, bitmaps, text blocks, and text fields. Levels in a SWF file are not composited on print output.
- The printer driver uses the HTML settings for dimension, scale, and alignment in the Publish Settings dialog box. Use these settings to control the print layout.
- The selected frames print as they appear in the movie clip symbol. Let users print a movie clip that is not visible in a browser by using the Actions panel to set the movie clip's `_visible` property to `false`. Changing the property of a movie clip with the `setProperty` action, tweening, or any transformation tool does not affect how a movie clip prints.
- For a movie clip to be printable, it must be on the Stage or pasteboard and it must be given an instance name.
- All elements must be fully loaded to print. Use the movie clip `_framesloaded` property to check whether the printable content is loaded. For more information, see `_framesloaded` (`MovieClip._framesloaded` property) in the *ActionScript 2.0 Language Reference* at www.adobe.com/go/learn_fl_cs4_as2lr_en.

Using the ActionScript 2.0 PrintJob class

The ActionScript PrintJob class, in addition to improving print functionality available with the `print()` method, lets you render dynamic content at runtime, prompt users with a single print dialog box, and print an unscaled document with proportions that map to the proportions of the content. This capability is useful for rendering and printing external dynamic content, such as database content and dynamic text.

Additionally, with properties populated by the `PrintJob.start()` method, your document can access your user's printer settings, such as page height, width, and orientation, and you can configure your document to dynamically format Flash content that is appropriate for the printer settings.

Note: When you use the PrintJob class, users must have Flash Player 7 or later.

Note: For information on printing from SWF files at runtime using ActionScript 2.0, see [Learning ActionScript 2.0 in Adobe Flash](#).

Build a print job

Because you are spooling a print job to the user's operating system between your calls to the `PrintJob.start()` and `PrintJob.send()` methods, and because the PrintJob functions might temporarily affect the Flash Player internal view of onscreen Flash content, implement print-specific activities only between your calls to `PrintJob.start()` and `PrintJob.send()`. For example, the Flash content should not interact with the user between `PrintJob.start()` and `PrintJob.send()`. Instead, complete formatting of your print job, add pages to the print job, and send the print job to the printer.

- 1 Create an instance of the print job object: `new PrintJob()`.
- 2 Start the print job and display the print dialog box for the operating system: `PrintJob.start()`.
- 3 Add pages to the print job (call once per page to add to the print job): `PrintJob.addPage()`.
- 4 Send the print job to the printer: `PrintJob.send()`.
- 5 Delete the print job: `delete PrintJob`.

The following example shows ActionScript code that creates a print job for a button:

```
myButton.onRelease = function()
{
    var my_pj = new PrintJob();
    var myResult = my_pj.start();
    if(myResult){
        myResult = my_pj.addPage (0, {xMin : 0, xMax: 400, yMin: 0,
            yMax: 400});
        myResult = my_pj.addPage ("myMovieClip", {xMin : 0, xMax: 400,
            yMin: 400, yMax: 800},{printAsBitmap:true}, 1);
        myResult = my_pj.addPage (1, null,{printAsBitmap:false}, 2);
        myResult = my_pj.addPage (0);
        my_pj.send();
    }
    delete my_pj;
}
```

Only one print job can run at any given time. A second print job cannot be created until one of the following events occurs with the previous print job:

- The print job is entirely successful and the `PrintJob.send()` method is called.
- The `PrintJob.start()` method returns a value of `false`.

- The `PrintJob.addPage()` method returns a value of `false`.
- The `delete PrintJob` method is called.

Starting a print job

Calling the ActionScript 2.0 `PrintJob.start()` method spools the print job to the user's operating system and prompts the user's operating system print dialog box to appear.

If the user selects an option to begin printing, the `PrintJob.start()` method returns a value of `true`. (The value is `false` if the user cancels the print job, in which case the script should call only `delete`). If successful, the `PrintJob.start()` method sets values for the `paperHeight`, `paperWidth`, `pageHeight`, `pageWidth`, and `orientation` properties.

Depending on the user's operating system, an additional dialog box might appear until spooling is complete and the `PrintJob.send` function is called: Make calls to `PrintJob.addPage()` and `PrintJob.send()` expeditiously. If ten seconds elapse between the `PrintJob.start()` function call and the `PrintJob.send()` function call, which sends the print job to the printer, Flash Player effectively calls `PrintJob.send()`, causing any pages that are added by using `PrintJob.addPage()` to be printed and spooling to stop.

When a new print job is constructed, the `PrintJob()` properties are initialized to 0. When `PrintJob.start()` is called, after the user selects the print option in the operating system print dialog box, Flash Player retrieves the print settings from the operating system. The `PrintJob.start()` function populates the following properties:

Property	Type	Unit	Notes
<code>PrintJob.paperHeight</code>	number	points	Overall paper height
<code>PrintJob.paperWidth</code>	number	points	Overall paper width
<code>PrintJob.pageHeight</code>	number	points	Height of actual printable area on the page; does not include any user-set margins
<code>PrintJob.pageWidth</code>	number	points	Width of actual printable area on the page; does not include any user-set margins
<code>PrintJob.orientation</code>	string	n/a	Portrait or landscape orientation

Adding pages to a print job

You add pages to your print job with the ActionScript 2.0 `PrintJob.addPage()` method. Although the method can include as many as four parameters, the only required parameter is `target/level`. The three optional parameters are `printArea`, `options`, and `frameNum`.

If you are not using a particular optional parameter but are using other optional parameters, use `NULL` in place of the excluded optional parameter.

With all four parameters, the function uses the following syntax:

```
MyPrintJob.addPage(target[,printArea:Object, options:Object, frameNum:Number]):boolean;
```

If you provide an invalid parameter, the print job uses default parameter values.

Each call to add a new page is unique, which lets you modify parameters without affecting previously set parameters. For example, you can specify that one page print as a bitmap image and another page print as a vector graphic. You can add as many new pages to your print job as the print job requires. One call to add a page equals one printed page.

Note: Any ActionScript code that needs to be called to change a resulting printout must run before the `PrintJob.addPage()` method is called. The ActionScript can, however, run before or after a new `PrintJob()` method. If a frame has a call to `PrintJob.addPage()`, the call itself does not guarantee that the ActionScript script on that frame will run when that frame is printed.

Specifying a target

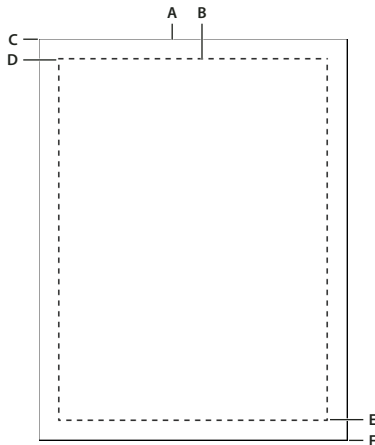
The ActionScript 2.0 `target` parameter can be either a number that represents a level (such as 0 for the `_root` document), or a string that represents the instance name of a movie clip ("myMovieClip").

Specifying a print area

The optional `printArea` parameter includes the following values:

```
{xMin:Number, xMax:Number, yMin:Number, yMax:Number}
```

The `xMin`, `xMax`, `yMin`, and `yMax` values represent screen pixels relative to the target level or movie clip registration point. The print area orientation is from the upper-left corner of the printable area on the page. If the print area is larger than the printable area on the page, the print data that exceeds the right and bottom edge of the page is clipped.



A. Paper rectangle B. Page rectangle C. (594,774) D. (576,756) E. (0,0) F. (-18,-18)

If you don't specify a print area, or if you specify an invalid print area, the print area defaults to the Stage area of the root document.

Scaling, points, and pixels

A print job using the `PrintJob` class prints Flash content, by default, without scaling it. For example, an object that is 144 pixels wide on screen prints as 144 points, or 2 inches wide.

One point equals one pixel. In the authoring tool, 72 pixels equals one inch; on paper, 72 points equals one inch.

To understand how Flash screen content maps to the printed page, it helps to understand screen and print units of measure. Pixels are a screen measurement and points are a print measurement. Both pixels and points equal 1/72 of an inch. A *twip* is 1/20 of a point and pixel.

The following list further illustrates the relationship between units of measure.

- 1 pixel = 20 twips
- 1 point = 20 twips
- 72 pixels = 1 inch

- 72 points = 1 inch
- 567 twips = 1 cm
- 1440 twips = 1 inch

To scale a movie clip before printing, set its `MovieClip.xscale` and `MovieClip.yscale` properties before calling this method, and set them back to their original values afterward. If you scale a movie clip and also pass a value for the `printArea` property, the pixel values passed to `printArea` reflect the original size of the movie clip. That is, if you set a movie clip's scale to 50% and specify a print area of 500 x 500 pixels, the content that prints is identical to the content that would print if you didn't scale the movie clip; however, it prints at half the size. For more information, see the `addPage` (`PrintJob.addPage` method) method of the `PrintJob` class in the [ActionScript 2.0 Language Reference](#).

Specifying printing as a vector image or bitmap graphic

The `options` parameter lets you specify whether to print as a vector graphic or bitmap image. When using this optional parameter, use the following syntax:

```
{printAsBitmap:boolean}
```

The default value is `false`, which represents a request for vector printing. Remember the following suggestions when determining which value to use:

- If the content that you're printing includes a bitmap image, specify that the print job print as a bitmap to include any transparency and color effects.
- Conversely, if the content does not include bitmap images, specify that the print job print as vector graphics to take advantage of the higher image quality.

Specifying a frame to print

The `frameNum` parameter lets you specify a frame to print. If you do not specify a frame number parameter, by default the current frame of the target or level that you specify as the first parameter when you add a page prints.

Print or delete a print job

- 1 To send a print job to the printer after using the ActionScript 2.0 `addPage()` calls, use the `PrintJob.send()` method, which causes Flash Player to stop spooling the print job so that the printer starts printing.
- 2 To delete the print job after you send the print job to a printer, use the `delete PrintJob()` ActionScript function to delete the `PrintJob` object, which frees memory.

Printing frames independent of the PrintJob class

The ActionScript 2.0 `PrintJob` class, available for Flash Player 7 and later, offers many advantages over the `print()` and `printAsBitmap()` methods for printing. However, to print targeting Flash Player 6 and earlier versions, back to Flash Player 4.0.25 (Windows) and 4.0.20 (Macintosh), use `print()` and `printAsBitmap()` functions and frame labels.

To set up printing from Flash Player independent of the `PrintJob` class, specify frames to print and set their print area.

Specifying a print area when not using the PrintJob object

By default, when frames are printed, the document file's Stage determines the print area. Any object that extends off the Stage is clipped and does not print. Loaded movies use their own Stage size for the print area, not the main SWF file's Stage size.

As an alternative to using a document's Stage size, set the following print areas:

- For either the Flash Player context menu or the ActionScript 2.0 `print()` function, designate the bounding box for SWF content as the print area for all frames by selecting an object in one frame as the bounding box. This option is useful, for example, to print a full-page data sheet from a web banner.
- The `print()` function lets you use the composite bounding box of all printable frames in a Timeline as the print area—for example, to print multiple frames that share a registration point. To use the composite bounding box, use the `bMax` parameter, as shown in the following example:

```
print ("myMovie", "bmax")
```

- The `print()` function lets you change the print area for each frame, scaling objects to fit the print area—for example, to have objects of different sizes in each frame fill the printed page. To change the bounding box per frame, use the `Frame` parameter in the Print action parameters, as shown in the following example:

```
print ("myMovie", "bframe")
```

- The `print()` function lets you designate the bounding box of a specific frame in a document as the print area for all printable frames in the document, as shown in the following example:

```
print ("myMovie", "bmovie")
```

You use the frame label `#b` to designate a frame to be used to designate the print area. The label `#b` must be on the same layer as a frame labeled `#p`.

For more information about `print()` function parameters, see “print function” in the [ActionScript 2.0 Language Reference](#).

Using the print() function when not using the PrintJob object

The basic syntax for the ActionScript 2.0 `print()` function, which you can associate with a button or other trigger in your document to activate printing, is as follows:

```
print (target, "Bounding box");
```

The `target` parameter specifies the location of the frames that print, and the `bounding box` parameter specifies the print area.

Assign a `print()` function to a button, frame, or movie clip in your document to let users print Flash content. If you assign a `print()` function to a frame, the action executes when the playhead reaches the designated frame.

The `print()` function lets you print frames in other movie clips in addition to the main Timeline. Each `print()` function sets only one Timeline for printing, but the action lets you specify any number of frames in the Timeline to print. If you attach more than one `print()` function to a single button or frame, the Print dialog box appears for each action executed.

See also “print function” in the [ActionScript 2.0 Language Reference](#).

ActionScript publish settings

Modify ActionScript publish settings

When you create a new FLA document, Flash asks you which version of ActionScript you want to use. You can change this setting if you decide later to write your scripts in a different version of ActionScript.

Note: The ActionScript 2.0 compiler can compile all ActionScript 1.0 code, except for the slash (/) syntax used to indicate movie clip paths (for example, `parentClip/testMC:varName= "hello world"`). To avoid this problem, either rewrite your code using dot (.) notation, or select the ActionScript 1.0 compiler.

- 1 Select File > Publish Settings and then select the Flash tab.
- 2 Select the ActionScript version from the pop-up menu.

Class files and configuration files

When you install Flash, several ActionScript-related configuration folders and files are placed on your system. If you modify these files to customize the authoring environment, back up the original files.

ActionScript classes folder Contains all of the built-in ActionScript 2.0 classes (AS files). Typical paths to this folder are as follows:

- Windows XP: Hard Disk\Documents and Settings\user\Local Settings\Application Data\Adobe\Flash CS4\language\Configuration\Classes
- Windows Vista: Hard Disk\Users\user\Local Settings\Application Data\Adobe\Flash CS4\language\Configuration\Classes
- Macintosh: Hard Disk/Users/user/Library/Application Support/Adobe/Flash CS4/language/Configuration/Classes

The Classes folder is organized into classes for Flash Player 7 (FP7), classes for Flash Player 8 (FP8), classes for Flash Player 9 (FP9), and the mx package, which is used in both players and in ASO files. For more information on the organization of this directory, see the Read Me file in the Classes folder.

Include class folder Contains all of the global ActionScript include files. Locations are as follows:

- Windows XP: Hard Disk\Documents and Settings\user\Local Settings\Application Data\Adobe\Flash CS4\language\Configuration\Include
- Windows Vista: Hard Disk\Users\user\Local Settings\Application Data\Adobe\Flash CS4\language\Configuration\Include
- Macintosh: Hard Disk/Users/user/Library/Application Support/Adobe/Flash CS4/language/Configuration/Include

ActionsPanel.xml configuration file Includes the configuration file for ActionScript code hinting. Separate files provide configuration for each version of ActionScript and Flash Lite, and for JavaScript. Locations are as follows:

- Windows XP: Hard Disk\Documents and Settings\user\Local Settings\Application Data\Adobe\Flash CS4\language\Configuration\ActionsPanel
- Windows Vista: Hard Disk\Users\user\Local Settings\Application Data\Adobe\Flash CS4\language\Configuration\ActionsPanel
- Macintosh: Hard Disk/Users/user/Library/Application Support/Adobe/Flash CS4/language/Configuration/ActionsPanel

AsColorSyntax.xml configuration file The configuration file for ActionScript code color syntax highlighting. Locations are as follows:

- Windows XP: Hard Disk\Documents and Settings\user\Local Settings\Application Data\Adobe\Flash CS4\language\Configuration\ActionsPanel\
- Windows Vista: Hard Disk\Users\user\Local Settings\Application Data\Adobe\Flash CS4\language\Configuration\ActionsPanel\

- Macintosh: Hard Disk/Users/user/Library/Application Support/Adobe/Flash CS4/language/Configuration/ActionsPanel

Declare an ActionScript 3.0 document class

When you use ActionScript 3.0, a SWF file may have a top-level class associated with it. This class is called the document class. When the SWF is loaded by Flash Player, an instance of this class is created to be the SWF file's top-level object. This object of a SWF file can be an instance of any custom class you choose.

For example, a SWF file that implements a calendar component can associate its top level with a Calendar class, with methods and properties appropriate to a calendar component. When the SWF is loaded, Flash Player creates an instance of this Calendar class.

- 1 Deselect all objects on the Stage and in the Timeline by clicking a blank area of the Stage. This displays the Document properties in the Property inspector.
- 2 Enter the filename of the ActionScript file for the class in the Document Class text box in the Property inspector. Do not include the .as filename extension.

Note: You can also enter the Document Class information in the Publish Settings dialog box.

Set the location of ActionScript files

To use an ActionScript class that you've defined, Flash must locate the external ActionScript files that contain the class definition. The list of folders in which Flash searches for class definitions is called the *classpath* for ActionScript 2.0 and the *source path* for ActionScript 3.0. Classpaths and source paths exist at the application (global) or document level. For more information about classpaths, see Classes in [Learning ActionScript 2.0 in Adobe Flash](#) or "Packages" in *Programming ActionScript 3.0*.

You can set the following ActionScript locations in Flash:

- ActionScript 2.0
 - Application level (available to all AS2 FLA files):
 - Classpath (set in ActionScript preferences)
 - Document level (available only to the FLA file that specifies this path):
 - Classpath (set in Publish Settings)
- ActionScript 3.0
 - Application level (available to all AS3 FLA files):
 - Source path (set in ActionScript preferences)
 - Library path (set in ActionScript preferences)
 - External library path (set in ActionScript preferences)
 - Document level (available only to the FLA file that specifies these paths):
 - Source path (set in Publish Settings)
 - Library path (set in Publish Settings)
 - External library path (set in Publish Settings)
 - Document class (set in Document Property inspector)

In ActionScript 3.0, there are two additional paths that you can set to specify the locations of ActionScript resources.

The *Library path* specifies the location of pre-compiled ActionScript code which resides in SWC files you have created. The FLA file that specifies this path loads every SWC file at the top level of this path and any other code resources that are specified within the SWC files themselves. If you use the Library path, be sure none of the compiled code in the SWC files is duplicated in uncompiled AS files in the Source path. The redundant code will slow down compilation of your SWF file.




The *External Library path* specifies the location of SWC files that contain code used for compiler error definitions. This path can be used to load external code resources in SWC format so that the classes within them can be used at runtime. When you compile a SWF, the SWC files in the External Library path are not added to the SWF file, but the compiler verifies that they are in the locations you specified. The External Library path is most often used for runtime shared libraries. For more information about runtime shared libraries, see “[Working with runtime shared assets](#)” on page 162. For more information about loading external code at runtime, see *Programming ActionScript 3.0*.

For each type of resource path, you can specify more than one path for Flash to look in. Resources found in any of the paths specified will be used.

When you add or modify a path, you can add absolute directory paths (for example, C:/my_classes) and relative directory paths (for example, ../my_classes or ".").

Set the classpath for ActionScript 2.0

To set the document-level classpath:

- 1 Select File > Publish Settings, and click Flash.
- 2 Verify that ActionScript 2.0 is selected in the ActionScript Version pop-up menu, and click Settings.
- 3 Specify the frame where the class definition should reside in the Export Frame for Classes text field.
- 4 To add paths to the classpath list, do any of the following:
 - To add a folder to the classpath, click the Browse to Path button , browse to the folder to add, and click OK.
 - To add a new line to the Classpath list, click the Add New Path  button. Double-click the new line, type a relative or absolute path, and click OK.
 - To edit an existing classpath folder, select the path in the Classpath list, click the Browse to Path button, browse to the folder to add, and click OK. Alternatively, double-click the path in the Classpath list, type the desired path, and click OK.
 - To delete a folder from the classpath, select the path in the Classpath list and click the Remove Selected Path button .




To set the application-level classpath:

- 1 Choose Edit Preferences (Windows) or Flash > Preferences (Macintosh) and click the ActionScript category.
- 2 Click the ActionScript 2.0 Settings button and add the path(s) to the Classpath list

Set the source path for ActionScript 3.0

To set the document-level source path:

- 1 Select File > Publish Settings, and click Flash.
- 2 Verify that ActionScript 3.0 is selected in the ActionScript Version pop-up menu, and click Settings. Your Flash Player version must be set to Flash Player 9 or later to use ActionScript 3.0.
- 3 Specify the frame where the class definition should reside in the Export Classes in Frame text field.

- 4 Specify the Errors settings. You can select Strict Mode and Warnings Mode. Strict Mode reports compiler warnings as errors, which means that compilation will not succeed if those types of errors exist. Warnings Mode reports extra warnings that are useful for discovering incompatibilities when updating ActionScript 2.0 code to ActionScript 3.0.
- 5 (Optional) Select Stage to automatically declare stage instances.
- 6 Specify ActionScript 3.0 or ECMAScript as the dialect to use. ActionScript 3.0 is recommended.
- 7 To add paths to the source path list, do any of the following:
 - To add a folder to the source path, click the Source path tab and then click the Browse To Path button , browse to the folder to add, and click OK.
 - To add a new line to the Source path list, click the Add New Path  button. Double-click the new line, type a relative or absolute path, and click OK.
 - To edit an existing Source path folder, select the path in the Source path list, click the Browse To Path button, browse to the folder to add, and click OK. Alternatively, double-click the path in the Source path list, type the desired path, and click OK.
 - To delete a folder from the source path, select the path in the Source path list and click the Remove From Path button .

To set the application-level source path:

- 1 Choose Edit Preferences (Windows) or Flash > Preferences (Macintosh) and click the ActionScript category.
- 2 Click the ActionScript 3.0 Settings button and add the path(s) to the Source path list.

Set the Library path for ActionScript 3.0 files

To set the document-level Library path, the procedure is similar to setting the Source path:

- 1 Choose File Publish Settings and click the Flash tab.
- 2 Make sure ActionScript 3.0 is specified in the Script menu and click Settings.
- 3 In the Advanced ActionScript 3.0 dialog box, click the Library path tab.
- 4 Add the library path to the Library path list. You can add folders or individual SWC files to the path list.

To set the Application-level Library path:

- 1 Choose Edit Preferences (Windows) or Flash > Preferences (Macintosh) and click the ActionScript category.
- 2 Click the ActionScript 3.0 Settings button and add the path(s) to the Library path list.

Set the External Library path for ActionScript 3.0 files

To set the document-level External Library path, the procedure is similar to setting the Source path:

- 1 Choose File Publish Settings and click the Flash tab.
- 2 Make sure ActionScript 3.0 is specified in the Script menu and click Settings.
- 3 In the Advanced ActionScript 3.0 dialog box, click the External Library path tab.
- 4 Add the library path to the External Library path list. You can add folders or individual SWC files to the path list.

To set the Application-level External Library path:

- 1 Choose Edit Preferences (Windows) or Flash > Preferences (Macintosh) and click the ActionScript category.
- 2 Click the ActionScript 3.0 Settings button and add the path(s) to the External Library path list.

Compiling ActionScript conditionally

You can use conditional compilation in ActionScript 3.0 in the same way that it has been used in C++ and other programming languages. For example, you can use conditional compilation to turn blocks of code throughout a project on or off, such as code that implements a certain feature or code used for debugging.

Using config constants that you define in the publish settings, you can specify whether certain lines of ActionScript code are compiled or not. Each constant takes the following form:

```
CONFIG::SAMPLE_CONSTANT
```

In this form, `CONFIG` is the config namespace and `SAMPLE_CONSTANT` is the constant that you will set to true or false in the publish settings. When the value of the constant is true, the line of code that follows the constant in ActionScript is compiled. When the value is false, the line of code that follows the constant is not compiled.

For example, the following function has 2 lines of code that are compiled only if the value of the constant that precedes them is set to true in the publish settings:

```
public function CondCompTest() {  
    CONFIG::COMPILE_FOR_AIR {  
        trace("This line of code will be compiled when COMPILE_FOR_AIR=true.");  
    }  
    CONFIG::COMPILE_FOR_BROWSERS {  
        trace("This line of code will be compiled when COMPILE_FOR_BROWSERS=true.");  
    }  
}
```

To define a config constant using the Publish Settings dialog box:

- 1 Choose File > Publish Settings.
- 2 In the Publish Settings dialog box, click the Flash tab.
- 3 Ensure that the value for Script is set to ActionScript 3.0 and click the Settings button next to the value.
- 4 In the Advanced ActionScript 3.0 Settings dialog box, click the Config Constants tab.
- 5 To add a constant, click the Add button.
- 6 Type the name of the constant you want to add. The default config namespace is `CONFIG` and the default constant name is `CONFIG_CONST`.

Note: The config namespace `CONFIG` is declared by the Flash compiler automatically. You can add your own config namespaces by entering them with a constant name in the publish settings and adding them to your ActionScript code using the following syntax:

```
config namespace MY_CONFIG;
```

- 7 Enter the value you want for the constant, either true or false. You change this value in order to turn on or off compilation of specific lines of code.

Customizing context menus in Flash documents

You can customize the standard context menu and the text-editing context menu that appears with SWF files in Flash Player 7 and later.

- The standard context menu appears when a user right-clicks (Windows) or Control-clicks (Macintosh) on a SWF file in Flash Player, in any area except an editable text field. You can add custom items to the menu, and hide any built-in items in the menu except Settings and Debugger.

- The editing context menu appears when a user right-clicks (Windows) or Control-clicks (Macintosh) in an editable text field in a SWF file in Flash Player. You can add custom items to this menu. You cannot hide any built-in items.

Note: *Flash Player also displays an error context menu when a user right-clicks (Windows) or Control-clicks (Macintosh) in Flash Player and no SWF file is loaded. You cannot customize this menu.*

Customize context menus in Flash Player 7 by using the `ContextMenu` and `ContextMenuItems` objects in ActionScript 2.0. For more information on using these objects, see `ContextMenu` in the [ActionScript 2.0 Language Reference](#).

Remember the following conditions when creating custom context menu items for Flash Player:

- Custom items are added to a context menu in the order in which they are created. You cannot modify this order after the items are created.
- You can specify the visibility and enabling of custom items.
- Custom context menu items are automatically encoded using Unicode UTF-8 text encoding.

Configuration folders installed with Flash

Flash places several configuration folders on your system when you install the application. The configuration folders organize files associated with the application into appropriate levels of user access. You may want to view the contents of these folders when you are working with ActionScript® or components. The configuration folders for Flash are as follows:

Application-level configuration folder

Because it is in the application level, non-administrative users do not have write access to this directory. Typical paths to this folder are as follows:

- In Microsoft Windows XP or Microsoft Windows Vista, browse to *boot drive*\Program Files\Adobe\Adobe Flash CS3\language\Configuration\.
- On the Macintosh, browse to *Macintosh HD*/Applications/Adobe Flash CS3/Configuration/.

First Run folder

This sibling to the application-level configuration folder facilitates sharing configuration files among users of the same computer. Folders and files in the First Run folder are automatically copied to the user-level configuration folder. Any new files placed in the First Run folder are copied to the user-level configuration folder when you start the application.

Typical paths to the First Run folder are as follows:

- In Windows XP or Vista, browse to *boot drive*\Program Files\Adobe\Adobe Flash CS3\language\First Run\.
- On the Macintosh, browse to *Macintosh HD*/Applications/Adobe Flash CS3/First Run/.

User-level configuration folder

Found in the user profile area, this folder is always writable by the current user. Typical paths to this folder are as follows:

- In Windows XP or Vista, browse to *boot drive*\Documents and Settings*username*\Local Settings\Application Data\Adobe\Flash CS3\language\Configuration.
- On the Macintosh, browse to *Macintosh HD*/Users/*username*/Library/Application Support/Adobe/Flash CS3/language/Configuration/.

All-user-level configuration folder

Found in the common user profile area, this folder is part of the standard Windows and Macintosh operating system installations and is shared by all users of a particular computer. The operating system makes available to all users of the computer any files placed in this folder. Typical paths to this folder are as follows:

- In Windows XP or Vista, browse to *boot drive*\Documents and Settings\All Users\Application Data\Adobe\Flash CS3\language\Configuration\.
- On the Macintosh, browse to *Macintosh HD*/Users/Shared/Application Support/Adobe/Flash CS3/language/Configuration/.

Restricted Users configuration folder

For users with restricted privileges on a workstation, typically, in a networked environment, only system administrators have administrative access to workstations. All other users are given restricted access, which usually means that these users can't write to application-level files (such as the Program Files directory in Windows or the Applications folder in Macintosh OS X).

Chapter 16: Publishing and Exporting

When you're ready to deliver Adobe® Flash® CS4 Professional content to an audience, you can publish it for playback or export it in a variety of formats. This chapter describes the various publishing and exporting options available with which to distribute your Flash content.

Publishing Flash documents

Publishing overview

By default, the Publish command creates a Flash SWF file and an HTML document that inserts your Flash content in a browser window. The Publish command also creates and copies detection files for Macromedia Flash 4 from Adobe and later. If you change publish settings, Flash saves the changes with the document. After you create a publish profile, export it to use in other documents or for others working on the same project to use.

Flash® Player 6 and later support Unicode text encoding. With Unicode support, users can view multilanguage text, regardless of the language that the operating system running the player uses.

You can publish the FLA file in alternative file formats—GIF, JPEG, PNG, and QuickTime®—with the HTML needed to display them in the browser window. Alternative formats allow a browser to show your SWF file animation and interactivity for users who don't have the targeted Adobe Flash Player installed. When you publish a Flash document (FLA file) in alternative file formats, the settings for each file format are stored with the FLA file.

You can export the FLA file in several formats, similar to publishing FLA files in alternative file formats, except that the settings for each file format are not stored with the FLA file.

Alternatively, create a custom HTML document with any HTML editor and include the tags required to display a SWF file.

To test how the SWF file works before you publish your SWF file, use Test Movie (Control > Test Movie) and Test Scene (Control > Test Scene).

See also

[“Using publish profiles”](#) on page 397

[“Publish settings”](#) on page 383

[“Configure a server for Flash Player”](#) on page 382

[“Creating multilanguage text”](#) on page 263

Playing Flash SWF files

The Flash SWF file format is for deploying Flash content.

You can play content in the following ways:

- In Internet browsers that are equipped with Flash Player
- With Flash Xtra in Director® and Authorware® from Adobe®
- With the Flash ActiveX control in Microsoft Office and other ActiveX hosts

- As part of a QuickTime video
- As a stand-alone application called a projector

The Flash SWF file format is an open standard that other applications support. For more information about Flash file formats, see www.adobe.com/go/flashplayer.

HTML documents

You need an HTML document to play a SWF file in a web browser and specify browser settings. To display a SWF file in a web browser, an HTML document must use the `object` and `embed` tags with the proper parameters.

Note: You can generate an HTML document using the correct `object` and `embed` tags using the Publish Settings dialog box, and selecting the HTML option. For more information, see “Specify publish settings for HTML documents” on page 384.

Flash can create the HTML document automatically when you publish a SWF file.

Publishing for mobile devices

Adobe® Flash® Lite® lets Flash users create engaging content for mobile phones using the ActionScript® scripting language, drawing tools, and templates. For detailed information on authoring for mobile devices, see *Developing Flash Lite Applications* and the Content Development Kits on the Mobile and Devices Development Center at www.adobe.com/go/devnet_devices.

Note: Depending on the mobile device for which you are developing, certain restrictions can apply as to which ActionScript commands and sound formats are supported. For more details, see *Mobile Articles on the Mobile and Devices Development Center*.

Adobe also provides Adobe Device Central, a new way to test content created with Adobe products on emulated mobile devices. When creating a new mobile document of any kind, start the creation process from Device Central. Device Central lets you select a target device from the beginning of the development process, and have a clear idea what a device’s limitations are.

Publishing secure Flash documents

Flash Player 8 and later contain the following features that help you ensure the security of your Flash documents:

Buffer overrun protection

Enabled automatically, this feature prevents the intentional misuse of external files in a Flash document to overwrite a user’s memory or insert destructive code such as a virus. This prevents a document from reading or writing data outside the document’s designated memory space on a user’s system.

Exact domain matching for sharing data between Flash documents

Flash Player 7 and later enforce a stricter security model than earlier versions. The security model changed in two primary ways between Flash Player 6 and Flash Player 7:

Exact domain matching Flash Player 6 lets SWF files from similar domains (for example, `www.adobe.com` and `store.adobe.com`) communicate freely with each other and with other documents. In Flash Player 7, the domain of the data to be accessed must match the data provider’s domain *exactly* for the domains to communicate.

HTTPS/HTTP restriction A SWF file that loads by using nonsecure (non-HTTPS) protocols cannot access content loaded by using a secure (HTTPS) protocol, even when both protocols are in exactly the same domain.

For more information about ensuring that content performs as expected with the new security model, see Understanding security in *Learning ActionScript 2.0 in Adobe Flash*.

Local and network playback security

Flash Player 8 and later include a security model that lets you determine the local and network playback security for SWF files that you publish. By default, SWF files are granted read access to local files and networks. However, a SWF file with local access cannot communicate with the network, and the SWF file cannot send files or information to any networks.

Allow SWF files to access network resources, letting the SWF file send and receive data. If you grant the SWF file access to network resources, local access is disabled, protecting information on the local computer from potentially being uploaded to the network.

To select the local or network playback security model for your published SWF files, use the Publish Settings dialog box.

See also

“Specify publish settings for SWF files” on page 383

Flash Player

Flash Player plays Flash content in the same way as it appears in a web browser or an ActiveX host application. Flash Player is installed with the Flash application. When you double-click Flash content, the operating system starts Flash Player, which then plays the SWF file. Use the player to make Flash content viewable for users who aren't using a web browser or an ActiveX host application.

To control Flash content in Flash Player, use menu commands and the `fscommand()` function. For more information, see Sending messages to and from Flash Player in *Learning ActionScript 2.0 in Adobe Flash*.

Use the Flash Player context menu to print Flash content frames.

❖ Do one of the following:

- To open a new or existing file, select File > New, or Open.
- To change your view of the application, select View > Magnification and make a selection.
- To control Flash content playback, select Control > Play, Rewind, or Loop Playback.

Reinstall Flash Player

If you are having trouble with your Flash Player installation, you can reinstall it.

- 1 Close your browser.
- 2 Remove any currently installed version of the player.

For instructions, see TechNote 14157 on the Adobe® Flash® Support Center at www.adobe.com/go/tn_14157.

- 3 To begin the installation, visit <http://www.adobe.com/go/getflashplayer>.

Follow the on-screen instructions to install the player.

You can also run one of the following installers in your Players folder. However, the installer on the Adobe website is usually more up to date than those in the Players folder.

- For the ActiveX control for Windows® (Internet Explorer or AOL), run the Install Flash Player 9 AX.exe file.
- For the plug-in for Windows (Firefox, Mozilla, Netscape, Safari, or Opera), run the Install Flash Player 9.exe file.

- For the plug-in for Macintosh® (AOL, Firefox, Internet Explorer for Macintosh, Netscape, Opera, or Safari), run Install Flash Player 10 (Mac OS 9.x) or Install Flash Player 10 OS X (Mac OS X.x).

Note: To verify the installation visit <http://www.adobe.com/shockwave/welcome/> from within your web browser.

Configure a server for Flash Player

For users to view your Flash content on the web, the web server must be properly configured to recognize SWF files.

Your server may already be configured properly. To test server configuration, see TechNote 4151 on the Adobe Flash Support Center at www.adobe.com/go/tn_4151.

Configuring a server establishes the appropriate Multipart Internet Mail Extension (MIME) types so that the server can identify files with the .swf extension as Flash files.

A browser that receives the correct MIME type can load the appropriate plug-in, control, or helper application to process and properly display the incoming data. If the MIME type is missing or not properly delivered by the server, the browser might display an error message or a blank window with a puzzle-piece icon.

- If your site is established through an Internet service provider (ISP), ask the ISP to add this MIME type to the server: application/x-shockwave-flash with the .swf extension.
- If you are administering your own server, see your web server documentation for instructions on adding or configuring MIME types.
- Corporate and enterprise system administrators can configure Flash to restrict Flash Player access to resources in the local file system. Create a security configuration file that limits Flash Player functionality on the local system.

The security configuration file is a text file placed in the same folder as the Flash Player installer. The Flash Player installer reads the configuration file during installation and follows its security directives. Flash Player uses the System object to expose the configuration file to ActionScript.

With the configuration file, disable Flash Player access to the camera or microphone, limit the amount of local storage Flash Player can use, control the auto-update feature, and block Flash Player from reading anything from the user's local hard disk.

For more information about security, see System in [ActionScript 2.0 Language Reference](#).

Adding MIME Types

When a web server accesses files, the server must properly identify the files as Flash content to display them. If the MIME type is missing or not properly delivered by the server, the browser can show error messages or a blank window with a puzzle-piece icon.

If your server is not properly configured, you (or your server's administrator) must add the SWF file MIME types to the server's configuration files and associate the following MIME types with the SWF file extensions:

- MIME type application/x-shockwave-flash has the .swf file extension.
- MIME type application/futuresplash has the .spl file extension.

If you are administering a server, consult your server software documentation for instructions on adding or configuring MIME types. If you are not administering a server, contact your Internet service provider, web master, or server administrator to add the MIME type information.

If your site is on a Mac OS server, you must also set the following parameters: Action: Binary; Type: SWFL; and Creator: SWF2.

Publish settings

Specify publish settings for SWF files

- 1 Select File > Publish Settings, click the Flash tab, and select a Player version from the Player pop-up menu. Not all Adobe® Flash® CS4 Professional features work in published SWF files that target Flash Player versions earlier than Flash Player 10. To specify Flash Player detection, click the HTML tab and select Detect Flash Version and enter Flash Player version to detect.
- 2 Select the ActionScript® version from the Script pop-up menu. If you select ActionScript 2.0 or 3.0 and you've created classes, click Settings to set the relative classpath to class files that differ from the path to default directories set in Preferences.
- 3 To control bitmap compression, adjust the JPEG Quality slider or enter a value. Lower image quality produces smaller files; higher image quality produces larger files. Try different settings to determine the best trade-off between size and quality; 100 provides the highest quality and least compression.

To make highly compressed JPEG images look smoother, select Enable JPEG Deblocking. This option reduces typical artifacts resulting from JPEG compression, such as the common appearance of 8x8-pixel blocking of the image. Some JPEG images may lose a small amount of detail when this option is selected.

- 4 To set the sample rate and compression for all streaming sounds or event sounds in the SWF file, click Set next to Audio Stream or Audio Event and select options as needed.

Note: A streaming sound plays as soon as enough data for the first few frames downloads; it is synchronized to the Timeline. An event sound does not play until it downloads completely, and it continues to play until explicitly stopped.

- 5 To override settings for individual sounds specified in the Sound section of the Property inspector, select Override Sound Settings. To create a smaller, low-fidelity version of a SWF file, select this option.

Note: If the Select Override Sound Settings option is deselected, Flash scans all stream sounds in the document (including sounds in imported video) and publishes all stream sounds at the highest individual setting. This can increase file size if one or more stream sounds has a high export setting.

- 6 To export sounds suitable for devices, including mobile devices, instead of the original library sound, select Export Device Sounds. Click OK.

- 7 To set SWF settings, select any of the following options:

Compress Movie (Default) Compresses the SWF file to reduce file size and download time. Most beneficial when a file is text- or ActionScript-intensive. A compressed file plays only in Flash Player 6 or later.

Include Hidden Layers (Default) Exports all hidden layers in the Flash document. Deselecting Export Hidden Layers prevents all layers (including layers nested inside movie clips) marked as hidden from being exported in the resulting SWF. This lets you easily test different versions of Flash documents by making layers invisible.

Include XMP metadata (Default) Exports all metadata entered in the File Info dialog box. Click the File Info button to open the dialog box. You can also open the File Info dialog box by choosing File > File Info. The metadata is viewable when the SWF file is selected in Adobe® Bridge.

Export SWC Exports a .swc file, which is used for distributing components. The .swc file contains a compiled clip, the component's ActionScript class file, and other files that describe the component.

- 8 To use advanced settings or enable debugging of the published Flash SWF file, select any of the following options:

Generate Size Report Generates a report listing the amount of data in the final Flash content by file.

Protect From Import Prevents others from importing a SWF file and converting it back into a FLA document. Lets you use password protection with your Flash SWF file.

Omit Trace Actions Causes Flash to ignore ActionScript `trace` statements in the current SWF file. When you select this option, information from `trace` statements does not appear in the Output panel. For more information, see [“Output panel overview”](#) on page 358.

Permit Debugging Activates the Debugger and allows remote debugging of a Flash SWF file. Lets you use password protection with your SWF file.

9 If you are using ActionScript 2.0, and selected either Permit Debugging or Protect From Import, enter a password in the Password text field. If you add a password, other users must enter the password before they can debug or import the SWF file. To remove the password, clear the Password text field. For more information on the Debugger, see [“Debugging ActionScript 1.0 and 2.0”](#) on page 350. If you are using ActionScript 3.0, see [“Debugging ActionScript 3.0”](#) on page 360.

10 Select the Flash security model to use from the Local Playback Security pop-up menu. Specify whether to grant your published SWF file local or network security access. Local Access Only lets the published SWF file interact with files and resources on the local system but not on the network. Access Network Only lets the published SWF file interact with files and resources on the network but not on the local system.

11 To enable the SWF file to use hardware acceleration, select one of the following options from the Hardware Acceleration menu:

Level 1 - Direct Direct mode improves playback performance by allowing Flash Player to draw directly on the screen instead of letting the browser do the drawing.

Level 2 - GPU In GPU mode, Flash Player utilizes the available computing power of the graphics card to perform video playback and compositing of layered graphics. This provides another level of performance benefit depending on the user's graphics hardware. Use this option when you expect that your audience will have high-end graphics cards.

If the playback system does not have sufficient hardware to enable acceleration, Flash Player reverts to normal drawing mode automatically. For best performance on web pages containing multiple SWF files, enable hardware acceleration for only one of the SWF files. Hardware acceleration is not used in Test Movie mode.

When you publish your SWF file, the HTML file that embeds it contains a `wmode` HTML parameter. Choosing Level 1 or Level 2 hardware acceleration sets the `wmode` HTML parameter to `direct` or `gpu` respectively. Turning on hardware acceleration overrides the Window Mode setting you may have chosen in the HTML tab of the Publish Settings dialog box, because it is also stored in the `wmode` parameter in the HTML file.

12 To set a maximum time that scripts can take to execute in the SWF file, enter a value for the Script Time Limit. Flash Player cancels execution of any scripts that exceed the limit.

See also

[“Using publish profiles”](#) on page 397

[“Sound”](#) on page 273

[“Using sounds in Flash Lite”](#) on page 277

Specify publish settings for HTML documents

Playing Flash content in a web browser requires an HTML document that activates the SWF file and specifies browser settings. The Publish command automatically generates this document from HTML parameters in a template document.

The template document can be any text file that contains the appropriate template variables—including a plain HTML file, a file that includes code for special interpreters such as ColdFusion® or Active Server Pages (ASP), or a template included with Flash.

To manually enter HTML parameters for Flash or customize a built-in template, use an HTML editor.

HTML parameters determine where the content appears in the window, the background color, the size of the SWF file, and so on, and set attributes for the `object` and `embed` tags. Change these and other settings in the HTML panel of the Publish Settings dialog box. Changing these settings overrides options you've set in the SWF file.

See also

[“Publishing overview”](#) on page 379

[“Configure a server for Flash Player”](#) on page 382

[“HTML publishing templates”](#) on page 406

[“Using publish profiles”](#) on page 397

[“Use device fonts”](#) on page 257

Specify the settings

- 1 Select File > Publish Settings and click Formats. The HTML file type is selected by default.
- 2 Use the default filename, which matches the name of your document, or enter a unique name, including the .html extension.
- 3 To show HTML settings and select an installed template to use, from the Template pop-up menu, click HTML. To show a description of the selected template, click Info. The default selection is Flash Only.
- 4 If you selected an HTML template other than Image Map or QuickTime, and on the Flash tab you set the Version to Flash Player 4 or later, select Flash Version Detection.

Note: Flash Version Detection configures your document to detect the version of Flash Player that the user has and sends the user to an alternative HTML page if the user does not have the targeted player.

- 5 Select a Dimensions option to set the values of the `width` and `height` attributes in the `object` and `embed` tags:

Match Movie (Default) Uses the size of the SWF file.

Pixels Enters the number of pixels for the width and height.

Percent Specifies the percentage of the browser window that the SWF file occupies.

- 6 To control the SWF file's playback and features, select Playback options:

Paused At Start Pauses the SWF file until a user clicks a button or selects Play from the shortcut menu. (Default) The option is deselected and the content begins to play as soon as it is loaded (the `PLAY` parameter is set to `true`).

Loop Repeats the content when it reaches the last frame. Deselect this option to stop the content when it reaches the last frame. (Default) The `LOOP` parameter is on.

Display Menu Shows a shortcut menu when users right-click (Windows) or Control-click (Macintosh) the SWF file. To show only About Flash in the shortcut menu, deselect this option. By default, this option is selected (the `MENU` parameter is set to `true`).

Device Font (Windows only) Substitutes anti-aliased (smooth-edged) system fonts for fonts not installed on the user's system. Using device fonts increases the legibility of type at small sizes and can decrease the SWF file's size. This option affects only SWF files that contain static text (text that you create when authoring a SWF file and that does not change when the content appears) set to display with device fonts.

- 7 To determine the trade-off between processing time and appearance, as described in the following list, select Quality options. These options set the `QUALITY` parameter's value in the `object` and `embed` tags.

Low Favors playback speed over appearance and does not use anti-aliasing.

Auto Low Emphasizes speed at first but improves appearance whenever possible. Playback begins with anti-aliasing turned off. If Flash Player detects that the processor can handle it, anti-aliasing is automatically turned on.

Auto High Emphasizes playback speed and appearance equally at first but sacrifices appearance for playback speed if necessary. Playback begins with anti-aliasing turned on. If the actual frame rate drops below the specified frame rate, anti-aliasing is turned off to improve playback speed. To emulate the View > Antialias setting, use this setting.

Medium Applies some anti-aliasing but does not smooth bitmaps. Medium produces a better quality than the Low setting but lower quality than the High setting.

High (Default) Favors appearance over playback speed and always uses anti-aliasing. If the SWF file does not contain animation, bitmaps are smoothed; if the SWF file contains animation, bitmaps are not smoothed.

Best Provides the best display quality and does not consider playback speed. All output is anti-aliased and bitmaps are always smoothed.

8 Select a Window Mode option, which controls the HTML `wmode` attribute in the `object` and `embed` tags. The window mode modifies the relationship of the content bounding box or virtual window with content in the HTML page as described in the following list:

Window (Default) Does not embed any window-related attributes in the `object` and `embed` tags. The background of the content is opaque and uses the HTML background color. The HTML code cannot render above or below the Flash content.

Opaque Windowless Sets the background of the Flash content to opaque, obscuring anything under the content. Lets HTML content appear above or on top of content.

Transparent Windowless Sets the background of the Flash content to transparent, allowing the HTML content to appear above and below the content. For browsers that support windowless modes, see “Parameters and attributes for object and embed tags” on page 387.

If you turn on Hardware Acceleration in the Flash tab of the Publish Settings dialog box, the Window Mode you select is ignored and defaults to Window.

Note: In some instances, complex rendering in Transparent Windowless mode can result in slower animation when the HTML images are also complex.

9 To position the SWF file window in the browser window, select one of the following HTML Alignment options:

Default Centers the content in the browser window and crops edges if the browser window is smaller than the application.

Left, Right, Top, Or Bottom Align SWF files along the corresponding edge of the browser window and crop the remaining three sides as needed.

10 To place the content within specified boundaries if you’ve changed the document’s original width and height, select a Scale option. The Scale option sets the `SCALE` parameter in the `object` and `embed` tags.

Default (Show All) Shows the entire document in the specified area without distortion while maintaining the original aspect ratio of the SWF files. Borders can appear on two sides of the application.

No Border Scales the document to fill the specified area and keeps the SWF file’s original aspect ratio without distortion, cropping the SWF file if needed.

Exact Fit Shows the entire document in the specified area without preserving the original aspect ratio, which can cause distortion.

No Scale Prevents the document from scaling when the Flash Player window is resized.

11 To set how the content is placed within the application window and how it is cropped, select the Flash Alignment option. This option sets the `SALIGN` parameter of the `object` and `embed` tags.

12 To show error messages if tag settings conflict—for example, if a template has code referring to an alternative image that was not specified—select Show Warning Messages.

13 To save the settings with the current file, click OK.

Parameters and attributes for object and embed tags

The following tag attributes and parameters describe the HTML code that the Publish command creates. Refer to this list as you write custom HTML to show Flash content. Unless noted, all items apply to both the `object` and `embed` tags. Optional entries are noted. Internet Explorer recognizes parameters used with the `object` tag; Netscape recognizes the `embed` tag. Attributes are used with both the `object` and `embed` tags. When you customize a template, you can substitute a template variable (identified in the Value section for each parameter in the following list) for the value.

Note: The attributes and parameters listed in this section are shown in lowercase to comply with the XHTML standard.

devicefont attribute/parameter (Optional) Specifies whether static text objects are rendered in device fonts, even if the Device Font option is not selected. This attribute applies when the necessary fonts are available from the operating system.

Value: `true` | `false`

Template variable: `$DE`

src attribute Specifies the name of the SWF file to be loaded. Applies to the `embed` tag only.

Value: `movieName.swf`

Template variable: `$MO`

movie parameter Specifies the name of the SWF file to be loaded. Applies to the `object` tag only.

Value: `movieName.swf`

Template variable: `$MO`

classid attribute Identifies the ActiveX control for the browser. The value must be entered exactly as shown. Applies to the `object` tag only.

Value: `clsid:d27c6b6e-ae6d-11cf-96b8-444553540000`

width attribute Specifies the width of the application either in pixels or as a percentage of the browser window.

Value: `n` or `n%`

Template variable: `$WI`

height attribute Specifies the height of the application either in pixels or as a percentage of the browser window.

Note: Because Flash applications are scalable, quality doesn't degrade at different sizes if the aspect ratio is maintained. (For example, the following sizes all have a 4:3 aspect ratio: 640 x 480 pixels, 320 x 240 pixels, and 240 x 180 pixels.)

Value: `n` or `n%`

Template variable: `$HE`

codebase attribute Identifies the location of the Flash Player ActiveX control so that the browser can automatically download it if it is not already installed. The value must be entered exactly as shown. Applies to the `object` tag only.

Value: `http://fpdownload.adobe.com/pub/shockwave/cabs/flash/swflash.cab#version=7,0,0,0`

pluginspage attribute Identifies the location of the Flash Player plug-in so that the user can download it if it is not already installed. The value must be entered exactly as shown. Applies to the `embed` tag only.

Value: `http://www.adobe.com/shockwave/download/index.cgi?P1_Prod_Version=ShockwaveFlash`

swliveconnect attribute (Optional) Specifies whether the browser should start Java™ when loading Flash Player for the first time. The default value is `false` if this attribute is omitted. If you use JavaScript and Flash on the same page, Java must be running for the `fscommand()` function to work. However, if you use JavaScript only for browser detection or another purpose unrelated to `fscommand()` actions, you can prevent Java from starting by setting `SWLIVECONNECT` to `false`. To force Java to start when you are not using JavaScript, explicitly set the `SWLIVECONNECT` attribute to `true`. Starting Java substantially increases the startup time for a SWF file; set this tag to `true` only when necessary. Applies to the `embed` tag only.

Use the `fscommand()` action to start Java from a stand-alone projector file.

Value: `true` | `false`

play attribute/parameter (Optional) Specifies whether the application begins playing immediately on loading in the web browser. If your Flash application is interactive, let the user initiate play by clicking a button or performing another task. In this case, set the `play` attribute to `false` to prevent the application from starting automatically. The default value is `true` if this attribute is omitted.

Value: `true` | `false`

Template variable: `$PL`

loop attribute/parameter (Optional) Specifies whether the content repeats indefinitely or stops when it reaches the last frame. The default value is `true` if this attribute is omitted.

Value: `true` | `false`

Template variable: `$LO`

quality attribute/parameter (Optional) Specifies the level of anti-aliasing to be used. Because anti-aliasing requires a faster processor to smooth each frame of the SWF file before it is rendered on the viewer's screen, select one of the following values based on whether your priority is speed or appearance:

- **Low** Favors playback speed over appearance and never uses anti-aliasing.
- **Autolow** Emphasizes speed at first but improves appearance whenever possible. Playback begins with anti-aliasing turned off. If Flash Player detects that the processor can handle it, anti-aliasing is turned on. Note: SWF files authored using ActionScript 3.0 do not recognize the `autolow` value.
- **Autohigh** Initially emphasizes playback speed and appearance equally, but sacrifices appearance for playback speed if necessary. Playback begins with anti-aliasing turned on. If the frame rate drops below the specified frame rate, anti-aliasing is turned off to improve playback speed. Use this setting to emulate the `Antialias` command (View > Preview Mode > Antialias).
- **Medium** Applies some anti-aliasing and does not smooth bitmaps. It produces a better quality than the Low setting but a lower quality than the High setting.
- **High** Favors appearance over playback speed and always applies anti-aliasing. If the SWF file does not contain animation, bitmaps are smoothed; if the SWF file has animation, bitmaps are not smoothed.
- **Best** Provides the best display quality and does not consider playback speed. All output is anti-aliased, and all bitmaps are smoothed.

The default value for `quality` is `high` if this attribute is omitted.

Value: `low` | `medium` | `high` | `autolow` | `autohigh` | `best`

Template variable: `$QU`

bgcolor attribute/parameter (Optional) Specifies the background color of the application. Use this attribute to override the background color setting that the SWF file specifies. This attribute does not affect the background color of the HTML page.

Value: #RRGGBB (hexadecimal RGB value)

Template variable: \$BG

scale attribute/parameter (Optional) Defines how the application is placed in the browser window when `width` and `height` values are percentages.

- **Showall (Default)** Makes the entire content visible in the specified area without distortion while maintaining the original aspect ratio of the application. Borders can appear on two sides of the application.
- **Noborder** Scales the content to fill the specified area, without distortion but possibly with some cropping, while maintaining the original aspect ratio of the application.
- **Exactfit** Makes the entire content visible in the specified area without trying to preserve the original aspect ratio. Distortion can occur.

The default value is `showall` if this attribute is omitted (and `width` and `height` values are percentages).

Value: `showall` | `noborder` | `exactfit`

Template variable: \$SC

align attribute Specifies the `align` value for the `object`, `embed`, and `img` tags and determines how the SWF file is positioned within the browser window.

- **Default** Centers the application in the browser window and crops edges if the browser window is smaller than the application.
- **L, R, T, and B** Align the application along the left, right, top, or bottom edge, respectively, of the browser window and crop the remaining three sides as needed.

Value: `Default` | `L` | `R` | `T` | `B`

Template variable: \$HA

salgn parameter (Optional) Specifies where a scaled SWF file is positioned in the area that the `width` and `height` settings define.

- **L, R, T, And B** Align the application along the left, right, top, or bottom edge, respectively, of the browser window and crop the remaining three sides as needed.
- **TL And TR** Align the application to the top-left and top-right corner, respectively, of the browser window and crop the bottom and remaining right or left side as needed.
- **BL And BR** Align the application to the bottom-left and bottom-right corner, respectively, of the browser window and crop the top and remaining right or left side as needed.

If this attribute is omitted, the content is centered in the browser window.

Value: `L` | `R` | `T` | `B` | `TL` | `TR` | `BL` | `BR`

Template variable: \$SA

base attribute (Optional) Specifies the base directory or URL used to resolve all relative path statements in the SWF file. This attribute is helpful when you keep SWF files in a different folder from your other files.

Value: base directory or URL

menu attribute or parameter (Optional) Specifies what type of menu appears when the viewer right-clicks (Windows) or Command-clicks (Macintosh) the application area in the browser.

- **true** shows the full menu, which gives the user several options to enhance or control playback.
- **false** shows a menu that contains only the About Adobe Flash Player 6 option and the Settings option.

The default value is `true` if this attribute is omitted.

Value: `true` | `false`

Template variable: `$ME`

wmode attribute or parameter (Optional) Lets you use the transparent Flash content, absolute positioning, and layering capabilities available in Internet Explorer 4.0. For a list of browsers this attribute/parameter supports, see “[Publishing Flash documents](#)” on page 379. The `wmode` parameter is also used for hardware acceleration in Flash Player 9 and later.

- **Window** Plays the application in its own rectangular window on a web page. Window indicates that the Flash application has no interaction with HTML layers and is always the top-most item.
- **Opaque** Makes the application hide everything behind it on the page.
- **Transparent** Makes the background of the HTML page show through all the transparent portions of the application and can slow animation performance.
- **Opaque windowless and Transparent windowless** Both interact with HTML layers, letting layers above the SWF file block out the application. Transparent allows transparency so that HTML layers below the SWF file might show through if a section of the SWF file has transparency; opaque does not.
- **Direct** Level 1 - Direct mode hardware acceleration is turned on. The other window mode settings apply only when hardware acceleration is turned off.
- **GPU** Level 2 - GPU mode hardware acceleration is turned on. The other window mode settings apply only when hardware acceleration is turned off.

For more information about hardware acceleration, see “[Specify publish settings for SWF files](#)” on page 383.

The default value is `Window` if this attribute is omitted. Applies to `object` only.

Value: `Window` | `Opaque` | `Transparent` | `Direct` | `GPU`

Template variable: `$WM`

allowscriptaccess attribute or parameter Use `allowscriptaccess` to let your Flash application communicate with the HTML page hosting it. The `fscommand()` and `getURL()` operations can cause JavaScript to use the permissions of the HTML page, which can be different from the permissions of your Flash application. This has important implications for cross-domain security.

- **always** Permits scripting operations at all times.
- **never** Forbids all scripting operations.
- **samedomain** Permits scripting operations only if the Flash application is from the same domain as the HTML page.

The default value that all HTML publish templates use is `samedomain`.

Value: `always` | `never` | `samedomain`

SeamlessTabbing parameter (Optional) Lets you set the ActiveX control to perform seamless tabbing, so that the user can tab out of a Flash application. This parameter works only in Windows with the Flash Player ActiveX control, version 7 and higher.

- **true** (or omitted) Sets the ActiveX control to perform seamless tabbing: After users tab through the Flash application, the next tab keypress moves the focus out of the Flash application and into the surrounding HTML content or to the browser status bar if nothing can have focus in the HTML following the Flash application.
- **false** Sets the ActiveX control to behave as it did in version 6 and earlier: After users tab through the Flash application, the next tab keypress wraps the focus around to the beginning of the Flash application. In this mode, you cannot use the tab key to advance the focus past the Flash application.

Value: true | false

Examples using object and embed tags

For `object`, four settings (`height`, `width`, `classid`, and `codebase`) are attributes that appear in the `object` tag; all others are parameters that appear in separate, named `param` tags, as shown in the following example:

```
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000" width="100"
height="100"
codebase="http://fpdownload.adobe.com/pub/shockwave/cabs/flash/swflash.cab#version=9,0,0,0">
<param name="movie" value="moviename.swf">
<param name="play" value="true">
<param name="loop" value="true">
<param name="quality" value="high">
</object>
```

For the `embed` tag, all settings (such as `height`, `width`, `quality`, and `loop`) are attributes that appear between the angle brackets of the opening `embed` tag, as shown in the following example:

```
<embed src="moviename.swf" width="100" height="100" play="true"
loop="true" quality="high"
pluginspage="http://www.adobe.com/shockwave/download/index.cgi?P1_Prod_Version=ShockwaveFlash">
</embed>
```

To use both tags, position the `embed` tag before the closing `object` tag, as shown in the following example:

```
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000" width="100"
height="100"
codebase="http://fpdownload.adobe.com/pub/shockwave/cabs/flash/swflash.cab#version=9,0,0,0">
<param name="movie" value="moviename.swf">
<param name="play" value="true">
<param name="loop" value="true">
<param name="quality" value="high">
<embed src="moviename.swf" width="100" height="100" play="true"
loop="true" quality="high"
pluginspage="http://www.adobe.com/shockwave/download/index.cgi?P1_Prod_Version=ShockwaveFlash">
</embed>
</object>
```

Note: If you use the `object` and `embed` tags, use identical values for each attribute or parameter to ensure consistent playback across browsers. The `swflash.cab#version=9,0,0,0` parameter is optional; only omit this parameter if you don't want to check for the version number.

Browsers that support windowless modes

Operating system	Internet Explorer	Netscape	Other
Macintosh OS X 10.1.5 and 10.2	5.1 and 5.2	7.0 and later	<ul style="list-style-type: none"> • Opera 6 or later • Mozilla 1.0 or later • AOL/CompuServe
Windows	5.0, 5.5, and 6.0	7.0 and later	<ul style="list-style-type: none"> • Opera 6 and later • Mozilla 1.0 and later • AOL/CompuServe

Specify publish settings for Flash Player detection

Flash Player detection is available only for publish settings set to Flash Player 4 or later, and for SWF files embedded in the Flash Only or Flash HTTPS templates.

***Note:** Flash Player 5 and later are installed on 98% of Internet-connected computers, making Flash Player detection a reasonable method to ensure that end users have the correct version of Flash installed to view your content.*

To detect the presence of Flash Player before allowing a browser to view SWF files that use the following templates, create a separate HTML page with its own SWF file to detect Flash Player before redirecting browsers to the HTML page that contains the Flash content.

The following HTML templates do not support Flash Player detection because the JavaScript in these templates conflicts with the JavaScript used to detect the Flash Player:

- Flash for PocketPC 2003
- Flash with AICC Tracking
- Flash with FSCommand
- Flash with Named Anchors
- Flash with SCORM Tracking

***Note:** Image Map and QuickTime HTML templates do not support Player detection because they do not embed the Flash Player.*

- 1 Select File > Publish Settings, and click HTML.
- 2 Select either the Flash Only or Flash HTTPS template from the Template pop-up menu. These templates support the single-page HTML detection kit. Either of these templates enables the Detect Flash Version check box and the version number text fields.
- 3 Select the Detect Flash Version check box. Your SWF file is embedded in a web page that includes Flash Player detection code. If the detection code finds an acceptable version of Flash Player installed on the end user's computer, the SWF file plays as designed.
- 4 (Optional) To specify precise revisions of Flash Player, use the Major Revision and Minor Revision text fields. For example, specify Flash Player version 7.0.2 if it provides a feature specific to displaying your SWF file.

When you publish your SWF file, Flash creates a single HTML page in which to embed the SWF file and the Flash Player detection code. If an end user does not have the version of Flash you've specified to view the SWF file, an HTML page appears with a link to download the latest version of Flash Player.

Specify publish settings for GIF files

Use GIF files to export drawings and simple animations for use in web pages. Standard GIF files are compressed bitmaps.

An animated GIF file (sometimes referred to as a GIF89a) offers a simple way to export short animation sequences. Flash optimizes an animated GIF file, storing only frame-to-frame changes.

Flash exports the first frame in the SWF file as a GIF file, unless you mark a different keyframe for export by entering the **#Static** frame label in the Property inspector. Flash exports all the frames in the current SWF file to an animated GIF file unless you specify a range of frames for export by entering the **#First** and **#Last** frame labels in the appropriate keyframes.

Flash can generate an image map for a GIF file to maintain URL links for buttons in the original document. Use the Property inspector to place the frame label **#Map** in the keyframe in which to create the image map. If you don't create a frame label, Flash creates an image map using the buttons in the last frame of the SWF file. Create an image map only if the `$IM` template variable is present in the template you select.

- 1 Select File > Publish Settings, click Formats, and select GIF Image.
- 2 For the GIF filename, use the default filename or enter a new filename with the .gif extension.
- 3 Click GIF.

Dimensions Enter values for width and height in pixels for the exported bitmap image, or select Match Movie to make the GIF the same size as the SWF file and maintain the aspect ratio of your original image.

Playback Determines whether Flash creates a still (Static) image or an animated GIF (Animation). If you select Animation, select Loop Continuously or enter the number of repetitions.

- 4 To specify a range of appearance settings for the exported GIF file, select one of the following options:

Optimize Colors Removes any unused colors from a GIF file's color table. This option reduces the file size without affecting image quality, but slightly increases the memory requirements. This option has no effect on an adaptive palette. (An adaptive palette analyzes the colors in the image and creates a unique color table for the selected GIF file.)

Interlace Incrementally shows the exported GIF file in a browser as it downloads. Lets the user see basic graphic content before the file completely downloads and can download the file faster over a slow network connection. Do not interlace an animated GIF image.

Smooth Applies anti-aliasing to an exported bitmap to produce a higher-quality bitmap image and improve text display quality. However, smoothing might cause a halo of gray pixels to appear around an anti-aliased image placed on a colored background, and it increases the GIF file size. Export an image without smoothing if a halo appears or if you're placing a GIF transparency on a multicolored background.

Dither Solids Applies dithering to solid colors as well as gradients.

Remove Gradients (Default is off) Converts all gradient fills in the SWF file to solid colors using the first color in the gradient. Gradients increase the size of a GIF file and are often poor quality. To prevent unexpected results, select the first color of your gradients carefully if you use this option.

- 5 To determine the transparency of the application's background and the way alpha settings are converted to GIF, select one of the following Transparent options:

Opaque Makes the background a solid color.

Transparent Makes the background transparent.

Alpha Sets partial transparency. Enter a Threshold value between 0 and 255. A lower value results in greater transparency. A value of 128 corresponds to 50% transparency.

6 To specify how pixels of available colors are combined to simulate colors not available in the current palette, select a Dither option. Dithering can improve color quality, but it increases the file size.

None Turns off dithering and replaces colors not in the basic color table with the solid color from the table that most closely approximates the specified color. Turning dithering off can result in smaller files but unsatisfactory colors.

Ordered Provides good-quality dithering with the smallest increase in file size.

Diffusion Provides the best-quality dithering but increases file size and processing time. Works only with the web 216-color palette selected.

7 To define the image's color palette, select one of the following Palette types:

Web 216 Uses the standard 216-color, web-safe palette to create the GIF image, for good image quality and the fastest processing on the server.

Adaptive Analyzes the colors in the image and creates a unique color table for the selected GIF file. Best for systems displaying thousands or millions of colors; it creates the most accurate color for the image but increases file size. To reduce the size of a GIF file with an adaptive palette, use the Max Colors option to decrease the number of colors in the palette.

Web Snap Adaptive Is the same as the Adaptive palette option except it converts similar colors to the web 216-color palette. The resulting color palette is optimized for the image, but when possible Flash uses colors from the web 216-color palette. This produces better colors for the image when the web 216-color palette is active on a 256-color system.

Custom Specifies a palette that you optimized for the selected image. The custom palette is processed at the same speed as the web 216-color palette. To use this option, know how to create and use custom palettes. To select a custom palette, click the Palette folder icon (the folder icon that appears at the end of the Palette text field), and select a palette file. Flash supports palettes saved in the ACT format that some graphics applications export.

8 To set the number of colors used in the GIF image, if you selected the Adaptive or Web Snap Adaptive palette, enter a value for Max Colors. A smaller number of colors can produce a smaller file but can degrade the colors in the image.

9 Click OK.

See also

[“Create an image map to substitute for a SWF file”](#) on page 410

[“Using publish profiles”](#) on page 397

[“Import and export color palettes”](#) on page 128

Specify publish settings for JPEG files

The JPEG format lets you save an image as a highly compressed, 24-bit bitmap. Generally, GIF format is better for exporting line art, and JPEG format is better for images with continuous tones, such as photographs, gradients, or embedded bitmaps.

Flash exports the first frame in the SWF file as a JPEG, unless you mark a different keyframe for export by entering the #`Static` frame label.

1 Select File > Publish Settings, click Formats, and select JPEG Image.

2 For the JPEG filename, either use the default filename, or enter a new filename with the .jpg extension.

3 Click the JPEG tab.

Dimensions Enter values for width and height in pixels for the exported bitmap image, or select Match Movie to make the JPEG image the same size as the Stage and maintain the aspect ratio of your original image.

Quality Drag the slider or enter a value to control the amount of JPEG file compression. The lower the image quality, the smaller the file size, and the reverse. To determine the best compromise between size and quality, try different settings.

***Note:** To change the object's compression setting, use the Bitmap Properties dialog box to set the bitmap export quality per object. The default compression option in the Bitmap Properties dialog box applies the Publish Settings JPEG Quality option.*

Progressive Show Progressive JPEG images incrementally in a web browser, which makes images appear faster when loading with a slow network connection. Similar to interlacing in GIF and PNG images.

4 Click OK.

See also

[“Using publish profiles”](#) on page 397

[“Set bitmap properties”](#) on page 87

Specify publish settings for PNG files

PNG is the only cross-platform bitmap format that supports transparency (an alpha channel). It is also the native file format for Adobe® Fireworks®.

Flash exports the first frame in the SWF file as a PNG file, unless you mark a different keyframe for export by entering the `#Static frame` label.

1 Select File > Publish Settings, click Formats, and select PNG Image.

2 For the PNG filename, either use the default filename, or enter a new filename with the .png extension.

3 Click PNG.

Dimensions Enter values for width and height in pixels for the exported bitmap image, or select Match Movie to make the PNG image the same size as the SWF file and maintain the aspect ratio of your original image.

Bit depth Set the number of bits per pixel and colors to use in creating the image. The higher the bit depth, the larger the file.

- 8-bits per channel (bpc) for a 256-color image
- 24-bpc for thousands of colors
- 24-bpc with Alpha for thousands of colors with transparency (32 bpc)

4 To specify appearance settings for the exported PNG, select from the following options:

Optimize Colors Removes any unused colors from a PNG file's color table, reducing the file size by 1000 to 1500 bytes without affecting image quality but increasing the memory requirements slightly. Has no effect on an adaptive palette.

Interlace Incrementally shows the exported PNG in a browser as it downloads. Lets the user see basic graphic content before the file completely downloads and might download the file faster over a slow network connection. Do not interlace an animated PNG file.

Smooth Applies anti-aliasing to an exported bitmap to produce a higher-quality bitmap image and improve text display quality. However, smoothing might cause a halo of gray pixels to appear around an anti-aliased image placed

on a colored background, and it increases the PNG file size. Export an image without smoothing if a halo appears or if you're placing a PNG transparency on a multicolored background.

Dither Solids Applies dithering to solid colors and gradients.

Remove Gradients (Default is off) Converts all gradient fills in the application to solid colors using the first color in the gradient. Gradients increase the size of a PNG and are often poor quality. To prevent unexpected results, select the first color of your gradients carefully if you use this option.

5 If you selected 8-bpc for Bit Depth, select a Dither option to specify how pixels of available colors are mixed to simulate colors not available in the current palette. Dithering can improve color quality, but it increases file size. Select from the following options:

None Turns off dithering and replaces colors not in the basic color table with the solid color from the table that most closely approximates the specified color. Turning dithering off can produce smaller files but unsatisfactory colors.

Ordered Provides good-quality dithering with the smallest increase in file size.

Diffusion Provides the best-quality dithering but increases file size and processing time. It also works only with the Web 216-color palette selected.

6 Select one of the following Palette Types to define the color palette for the PNG image:

Web 216 Uses the standard 216-color, web-safe palette to create the PNG image, for good image quality and the fastest processing on the server.

Adaptive Analyzes the colors in the image and creates a unique color table for the selected PNG file. Best for systems showing thousands or millions of colors; it creates the most accurate color for the image but results in a file size larger than a PNG created with the web-safe 216-color palette.

Web Snap Adaptive Is the same as the Adaptive palette option except that it converts colors similar to the web-safe 216-color palette. The resulting color palette is optimized for the image, but when possible, Flash uses colors from the web-safe 216-color palette. This produces better colors for the image when the web-safe 216-color palette is active on a 256-color system. To reduce the size of a PNG file with an adaptive palette, use the Max Colors option to decrease the number of palette colors.

Custom Specifies a palette that you optimized for the selected image. The custom palette is processed at the same speed as the web-safe 216-color palette. To use this option, know how to create and use custom palettes. To select a custom palette, click the Palette folder icon (the folder icon that appears at the end of the Palette text field), and select a palette file. Flash supports palettes saved in the ACT format that leading graphics applications export.

7 If you selected the Adaptive or Web Snap Adaptive palette, enter a value for Max Colors to set the number of colors used in the PNG image. A smaller number of colors can produce a smaller file but might degrade the colors in the image.

8 To select a line-by-line filtering method to make the PNG file more compressible and experiment with the different options for a particular image, select one of the following Filter options:

None Turns off filtering.

Sub Transmits the difference between each byte and the value of the corresponding byte of the previous pixel.

Up Transmits the difference between each byte and the value of the corresponding byte of the pixel immediately above.

Average Uses the average of the two neighboring pixels (left and above) to predict the value of a pixel.

Path Computes a simple linear function of the three neighboring pixels (left, above, upper left), and selects the neighboring pixel closest to the computed value as a predictor of the color.

Adaptive Analyzes the colors in the image and creates a unique color table for the selected PNG file. Best for systems showing thousands or millions of colors; it creates the most accurate color for the image but results in a file size larger than a PNG created with the web 216-color palette. Reduce the size of a PNG created with an adaptive palette by decreasing the number of colors in the palette.

9 Click OK.

See also

[“Using publish profiles”](#) on page 397

[“Import and export color palettes”](#) on page 128

Preview the publishing format and settings

The Publish Preview command exports the file and opens the preview in the default browser. If you preview a QuickTime video, Publish Preview starts the QuickTime video Player. If you preview a projector, Flash starts the projector.

❖ Select File > Publish Preview, and select the file format to preview.

Using the current Publish Settings values, Flash creates a file of the specified type in the same location as the FLA file. This file remains in this location until you overwrite or delete it.

See also

[“Publishing overview”](#) on page 379

Using publish profiles

Publish profiles let you:

- Save a publish settings configuration, export it, and import the publish profile to other documents or for others to use.
- Import publish profiles to use in your document.
- Create profiles to publish in several media formats.
- Create a publish profile for in-house use that differs from the way you’d publish the files for a client.
- Create a standard publish profile for your company to ensure files are published uniformly.

Publish profiles are saved at the document rather than application level.



See also

[“Publishing overview”](#) on page 379


Create a publish profile

- 1 In the Publish Settings dialog box, click the Create New Profile button **+**.
- 2 Name the publish profile, and click OK.
- 3 Specify the publish settings for your document, and click OK.

Duplicate, modify, or delete a publish profile


- ❖ From the Current Profile pop-up menu (File > Publish Settings), select the publish profile to use:
 - To create a duplicate profile, click the Duplicate Profile button . Enter the profile name in the Duplicate Name text field, and click OK.
 - To modify a publish profile, specify the new publish settings for your document, and click OK.
 - To delete a publish profile, click the Delete Profile button , and click OK.

Export a publish profile

- 1 From the Current Profile pop-up menu (File > Publish Settings), select the publish profile to export.
- 2 Click the Import/Export Profile button , and select Export. Export a publish profile as an XML file for import into other documents.
- 3 Either accept the default location in which to save the publish profile or browse to a new location, and click Save.

Import a publish profile

Other users can create and export publish profiles, which you can import and select as a publish settings option.

- 1 Select File > Publish Settings, click Import/Export Profile , and select Import.
- 2 Browse to the publish profile XML file, and click Open.

Publishing for Adobe AIR

Adobe® AIR™, a new cross-operating system runtime allows you to leverage local desktop resources and data to deliver even more personal, engaging experiences. Using the same skills you use to deliver to the Adobe Flash Player, your content can now reach even more audiences across more devices – web, mobile, and now the desktop.

About Adobe AIR

Adobe® AIR™ is a cross-operating system runtime that allows you to leverage your existing web development skills (Adobe® Flash® CS3 Professional, Adobe® Flex™, HTML, JavaScript®, Ajax) to build and deploy Rich Internet Applications (RIAs) to the desktop. AIR enables you to work in familiar environments, to leverage the tools and approaches you find most comfortable, and by supporting Flash, Flex, HTML, JavaScript, and Ajax, to build the best possible experience that meets your needs.

Users interact with AIR applications in the same way that they interact with native desktop applications. The runtime is installed once on the user's computer, and then AIR applications are installed and run just like any other desktop application. The runtime provides a consistent cross-operating system platform and framework for deploying applications and therefore eliminates cross-browser testing by ensuring consistent functionality and interactions across desktops. Instead of developing for a specific operating system, you target the runtime.

AIR dramatically changes how applications can be created, deployed, and experienced. You gain more creative control and can extend your Flash, Flex, HTML, and Ajax-based applications to the desktop, without learning traditional desktop development technologies.

For information on developing Adobe AIR™ applications, see [Developing Adobe AIR 1.5 Applications with Adobe Flash CS4 Professional](#).

Create an Adobe AIR file

You can create Flash File (Adobe AIR) documents using the Flash Welcome screen, or the File > New command, or you can create a Flash File (ActionScript® 3.0) and convert it to an Adobe AIR file through the Publish Settings dialog box.

To create an Adobe AIR file, do one of the following:

- Start Flash. The Welcome screen appears. If you have already started Flash, close any open documents to return to the Welcome screen. In the Welcome screen, select Flash File (Adobe AIR).

Note: If you've disabled the Flash Welcome screen, you can display it again by selecting Edit > Preferences and selecting Welcome Screen from the On Launch pop-up menu in the General category.

- Choose File > New and select Flash File (Adobe AIR) and click OK.
- Open an existing Flash file and convert it to an AIR file by selecting Adobe AIR from the Player menu in the Flash tab of the Publish Settings dialog box (File > Publish Settings).

By default, AIR files are set to use ActionScript 3.0. You can create an AIR file from an ActionScript 2.0 FLA file, but the file will not be able to use any of the AIR-specific APIs, which are all ActionScript 3.0. This can be useful for converting older Flash content into AIR applications.

Note: If you save a Flash CS4 AIR file in Flash CS3 format, set the Player version to AIR 1.0 manually in the Publish Settings dialog box when opening the file in Flash CS3. Flash CS3 only supports publishing to AIR 1.0.

Preview or publish an Adobe AIR application

You can preview a Flash AIR SWF file as it would appear in the AIR application window. Previewing is useful when you want to see what the visible aspects of the application look like without packaging and installing the application.

- 1 Make sure you've set the Player setting in the Flash tab of the Publish Settings dialog box to Adobe AIR.
- 2 Select Control > Test Movie or press Control+Enter.

If you have not set application settings through the AIR - Application And Installer Settings dialog box, Flash generates a default application descriptor file (*swfname-app.xml*) for you in the same folder where the SWF file is written. If you have set application settings using the AIR - Application And Installer Settings dialog box, the application descriptor file reflects those settings.

To publish an AIR file, do one of the following:

- Click the Publish button in the Publish Settings dialog box.
- Click the Publish AIR File button in the AIR - Application & Installer Settings dialog box.
- Choose File > Publish.
- Choose File > Publish Preview.

When you Publish an AIR file, Flash creates a SWF file and XML application descriptor file and packages copies of both, along with any other files you have added to your application, into an AIR installer file (*swfname.air*).

Creating AIR application and installer files

After you've finished developing your application, specify the settings for the AIR application descriptor and installer files required to deploy it. Flash creates the descriptor and installer files along with the SWF file when you publish an AIR file.

You specify the settings for these files in the AIR - Application & Installer Settings dialog box. Once you have created an AIR file, this dialog box can be opened from either the document Property inspector or the Player menu Settings button in the Flash tab of the Publish Settings dialog box.

Create the Adobe AIR application and installer files

- 1 In Flash, open the FLA file or set of files that make up your Adobe AIR application.
- 2 Save the Adobe AIR FLA file before you open the AIR - Application & Installer Settings dialog box.
- 3 Select File > AIR Settings.
- 4 Complete the AIR - Application & Installer Settings dialog box, and then click Publish AIR File.

When you click the Publish AIR File button, the following files are packaged: the SWF file, the application descriptor file, the application icon files, and the files listed in the Included Files text box. If you have not already created a digital certificate, Flash displays the Digital Signature dialog box when you click the Publish AIR File button.

The AIR - Application And Installer Settings dialog box is divided into two sections: Application Settings and Installer Settings. For more information on these settings, see the following sections.

Application settings

The Application settings section of the AIR - Application And Installer Settings dialog box contains the following options:

File Name The name of the main file of the application. Defaults to the name of the FLA file.

Name The name used by the AIR application installer to generate the application filename and the application folder. The name must contain only valid characters for filenames or folder names. Defaults to the name of the SWF file.

Version Optional. Specifies a version number for your application. Defaults to 1.0.

ID Identifies your application with a unique ID. You can change the default ID if you prefer. Do not use spaces or special characters in the ID. The only valid characters are 0-9, a-z, A-Z, . (dot), and - (dash), from 1 to 212 characters in length. Defaults to `com.adobe.example.applicationName`.

Description Optional. Lets you enter a description of the application to display in the installer window when the user installs the application. Defaults to blank.

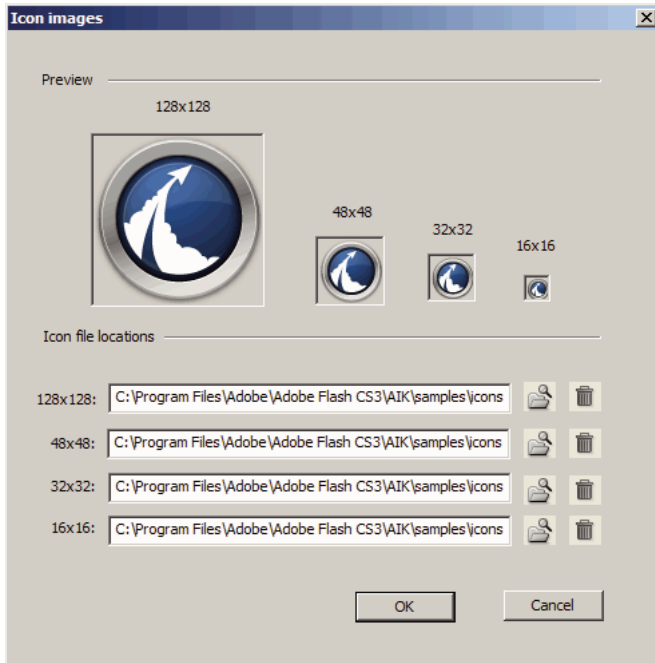
Copyright Optional. Lets you enter a copyright notice. Defaults to blank.

Window Style Specifies what window style (or chrome) to use for the user interface when the user runs the application on their computer. You can specify System Chrome (the default), which refers to the standard window visual style that the operating system uses. You can also specify Custom Chrome (opaque) or Custom Chrome (transparent). To display your application without the system chrome, select None. System Chrome surrounds the application with the operating-system standard window control. Custom Chrome (opaque) eliminates the standard system chrome and lets you create a chrome of your own for the application. (You build the custom chrome directly in the FLA file.) Custom Chrome (transparent) is like Custom Chrome (opaque), but it adds transparent capabilities to the edges of the page. These capabilities allow for application windows that are not square or rectangular in shape.

Icon Optional. Lets you specify an icon for the application. The icon is shown after you install the application and run it in the Adobe AIR runtime. You can specify four different sizes for the icon (128, 48, 32, and 16 pixels) to allow for the different views in which the icon appears. For example, the icon can appear in the file browser in thumbnail, detail, and tile views. It can also appear as a desktop icon and in the title of the AIR application window, as well as in other places.

The icon image defaults to a sample AIR application icon if no other icon files are specified.

To specify an icon, click the Select Icon Images button in the AIR - Application And Installer Settings dialog box. In the Icon images dialog box that appears, click the folder for each icon size and select the icon file to use. The files must be in PNG (Portable Network Graphics) format.



The Icon Images dialog box with the default Adobe AIR application icons

If you do specify an image, it must be the exact size (either 128x128, 48x48, 32x32, or 16x16). If you do not supply an image for a particular icon size, Adobe AIR scales one of the supplied images to create the missing icon image.

Advanced Settings

The Advanced Settings button in the AIR - Application And Installer Settings dialog box opens a separate dialog box that allows you to specify additional settings for the application descriptor file.

The Advanced Settings dialog box lets you specify any associated file types that your AIR application should handle. For example, if you wanted your application to be the principal application for handling HTML files, you would specify that in the Associated File Types text box.

You can also specify settings for the following aspects of the application:

- The size and placement of the initial window
- The folder in which the application is installed
- The Program menu folder in which to place the application.

The dialog box has the following options:

Associated file types Lets you specify associated file types that the AIR application will handle. Click the Plus (+) button to add a new file type to the text box. Clicking the Plus button displays the File Type Settings dialog box. Clicking the Minus (-) button removes an item that is selected in the text box. Clicking the Pencil button displays the File Type Settings dialog box and allows you to edit an item that you've selected in the text box. By default, the Minus (-) and Pencil buttons are dimmed. Selecting an item in the text box enables the Minus (-) and Pencil buttons, allowing you to remove or edit the item. The default value in the text box is None.

Initial window settings Lets you specify size and placement settings for the initial application window.

- **Width:** Specifies the initial width of the window in pixels. The value is blank by default.
- **Height:** Specifies the initial height of the window in pixels. The value is blank by default.
- **X:** Specifies the initial horizontal position of the window in pixels. The value is blank by default.
- **Y:** Specifies the initial vertical position of the window in pixels. The value is blank by default.
- **Maximum Width and Maximum Height:** Specify the maximum size of the window in pixels. These values are blank by default.
- **Minimum Width and Minimum Height:** Specify the minimum size of the window in pixels. These values are blank by default.
- **Maximizable:** Lets you specify whether the user can maximize the window. This option is selected (or true) by default.
- **Minimizable:** Lets you specify whether the user can minimize the window. This option is selected (or true) by default.
- **Resizable:** Lets you specify whether the user can resize the window. If this option is not selected, Maximum Width, Maximum Height, Minimum Width, and Minimum Height are dimmed. This option is selected (or true) by default.
- **Visible:** Lets you specify whether the application window is visible initially. The option is selected (or true) by default.

Other Settings Lets you specify the following additional information regarding the installation:

- **Install Folder:** Specifies the folder in which the application is installed.
- **Program Menu Folder (Windows only):** Specifies the name of the program menu folder for the application.
- **Use Custom UI for Updates:** Specifies what happens when a user opens an AIR installer file for an application that's already installed. By default, AIR displays a dialog box that allows the user to update the installed version with the version in the AIR file. If you don't want the user to make that decision and you want the application to have complete control over its updates, select this option. Selecting this option overrides the default behavior and gives the application control over its own updates.

File type settings

Flash displays the File Type Settings dialog box if you click the Plus (+) button or the Pencil button in the Associated File Types section of the Advanced Settings dialog box to add or edit associated file types for the application.

The only two required fields in this dialog box are Name and Extension. If you click OK and either of those fields is blank, Flash displays an error dialog box.

You can specify the following settings for an associated file type:

Name The name of the file type (for example, Hypertext Markup Language, Text File, or Example).

Extension The filename extension (for example, html, txt, or xml), up to 39 basic alphanumeric characters, (A-Za-z0-9), and without a leading period.

Description Optional. A description of the file type (for example, Adobe Video File).

Content type Optional. Specifies the MIME type for the file.

File Type Icon Settings Optional. Lets you specify an icon that's associated with the file type. You can specify four different sizes for the icon (128x128, 48x48, 32x32, and 16x16 pixels) to allow for the different views in which the icon appears. For example, the icon can appear in the file browser in thumbnail, detail, and tile views.

If you specify an image, it must be of the size that you specify. If you do not specify a file for a particular size, AIR uses the image of the closest size and scales it to fit for the given occurrence.

To specify an icon, either click the folder for the icon size and select an icon file to use or enter the path and filename for the icon file in the text box next to the prompt. The icon file must be in PNG format.

After a new file type is created, it is shown in the File Type list box in the Advanced Settings dialog box.

Application descriptor file settings

The application settings that you specify in the AIR - Application & Installer Settings dialog box are saved in the *application_name-app.xml* file when you publish your AIR application. You have the option, however, of indicating to Flash that you want to use a custom application descriptor file.

Use Custom Application Descriptor File Lets you browse to a custom application descriptor file. If you select Use Custom Application Descriptor File, the Application Settings section of the dialog box is dimmed. To specify the location of the custom application descriptor file, either enter it in the text field below Use Custom Application Descriptor File or click the folder icon and browse to the location. For more information on understanding and editing the application descriptor file, see *Creating a custom application descriptor file in the Developing AIR Applications* guide.

Installer settings

The second section of the AIR - Application And Installer Settings dialog box contains settings that pertain to installing the application.

Digital Signature All Adobe AIR applications must be signed to be installed on another system. For information about assigning a digital signature to a Flash Adobe AIR application, see “[Signing your application](#)” on page 405.

Destination Specifies where to save the AIR file. The default location is the directory where you saved the FLA file. Click the folder icon to select a different location. The default package name is the application name with the .air file extension.

Included Files/Folders Specifies which additional files and folders to include in your application package. Click the Plus (+) button to add files, and the folder button to add folders. To delete a file or folder from your list, select the file or folder and click the Minus (-) button.

By default, the application descriptor file and the main SWF file are automatically added to the package list. The package list shows these files even if you have not yet published the Adobe AIR FLA file. The package list displays the files and folders in a flat structure. Files in a folder are not listed, and full path names to files are shown but are truncated if necessary.

Icon files are not included in the list. When Flash packages the files, it copies the icon files to a temporary folder that is relative to the location of the SWF file. Flash deletes the folder after packaging is complete.

Failure to create application and installer files

The application and installer files fail to be created in the following instances:

- The application ID string has an incorrect length or contains invalid characters. The application ID string can be from 1 to 212 characters and can include the following characters: 0-9, a-z, A-Z, . (dot), - (hyphen).
- Files in the Included Files list do not exist.
- The sizes of custom icon files are incorrect.
- The AIR destination folder does not have write access.

- You have not signed the application or have not specified that it is an Adobe AIRI application that will be signed later.

Creating a custom application descriptor file

The application descriptor file is an XML file that you can edit with a text editor. Flash creates the descriptor file based on the settings you choose in the AIR - Application & Installer Settings dialog box. You can also choose to create your own descriptor file or edit the one that Flash creates.

To create a custom application descriptor file, edit the values to specify the settings you want. The default values are shown here:

- `id = com.adobe.example.swfname`
- `version = 1.0`
- `filename = swfname`
- `description = blank`
- `name = swfname`
- `copyright = blank`
- `initialWindow`
 - `content = swfname.swf`
 - `systemChrome = standard, type = normal`
 - `transparent = false`
 - `visible = true`
- `icon`
 - `image128x128 = icons/AIRApp_128.png`
 - `image48x48 = icons/AIRApp_48.png`
 - `image32x32 = icons/AIRApp_32.png`
 - `image16x16 = icons/AIRApp_16.png`
- `customUpdateUI = false`
- `allowBrowserInvocation = false`

You can customize the `description` and `name` properties to include languages other than English.

To provide a name or `description` for additional languages:

- 1 Delete the default `name` or `description` element.
- 2 Replace it with the example `name` or `description` element provided in the default descriptor file.

The example `name` element looks like this:

```
<!-- To localize the name, use the following format for the name element.

<name>

<text xml:lang="en">English App name goes here</text>

<text xml:lang="fr">French App name goes here</text>

<text xml:lang="ja">Japanese App name goes here</text>
```

</name>

-->

The `xml:lang` attribute for each `text` element specifies a language code, as defined in RFC4646 (<http://www.ietf.org/rfc/rfc4646.txt>).

- 3 When you are finished adding or removing `text` elements, remove the comment indicators (`<!--` and `-->`) from around the `name` element.

For more information on the application descriptor file and its properties, see Setting AIR application properties in the *Developing AIR Applications* guide at www.adobe.com/go/go/learn_air_en.

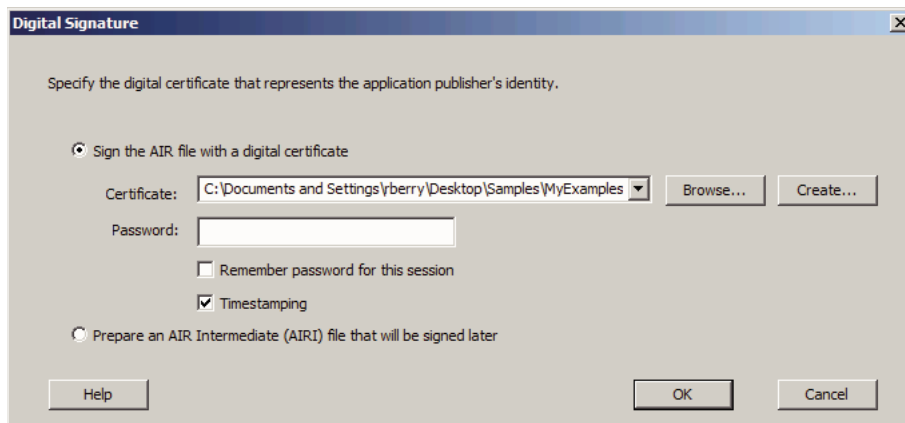
Signing your application

All Adobe AIR applications must be signed to be installed on another system. Flash provides the ability, however, to create unsigned Adobe AIR installer files so that the application can be signed later. These unsigned Adobe AIR installer files are called an AIRI (AIR Intermediate) package. This capability provides for cases in which the certificate is on a different machine or signing is handled separately from application development.

Sign an Adobe AIR application with a pre-purchased digital certificate from a root certificate authority

- 1 Click the Digital Signature Set button in the AIR - Application And Installer Settings dialog box. The Digital Signature dialog box opens.

This dialog box has two radio buttons that allow you to either sign your Adobe AIR application with a digital certificate or prepare an AIRI package. If you sign your AIR application, you can either use a digital certificate granted by a root certificate authority or create a self-signed certificate. A self-signed certificate is easy to create but is not as trustworthy as a certificate granted by a root certificate authority.



Digital Signature dialog box for signing an AIR application

- 2 Select a certificate file from the pop-up menu or click the Browse button to locate a certificate file.
- 3 Select the certificate.
- 4 Enter a password.
- 5 Click OK.

For more information on signing your AIR application, see Digitally signing an AIR file in the *Developing AIR Applications* guide.

Create a self-signed digital certificate

- 1 Click the Create button. The Self-Signed Digital Certificate dialog box opens.
- 2 Complete the entries for Publisher Name, Organization Unit, Organization Name, Country, Password, and Confirm Password. For Country, you can select from the menu or enter a 2-letter country code that does not appear in the menu. For a list of valid country codes, see http://www.iso.org/iso/country_codes.
- 3 Specify the type of certificate.
The Type option refers to the level of security that the certificate carries: 1024-RSA uses a 1024-bit key (less secure), and 2048-RSA uses a 2048-bit key (more secure).
- 4 Save the information in a certificate file by completing the Save As entry or clicking the Browse button to browse to a folder location.
- 5 Click OK.
- 6 In the Digital Signature dialog box, enter the password you assigned in the second step of this procedure and click OK. After you have set a digital certificate, the Set button changes to a Change button.

To have Flash remember the password you used for this session, click Remember Password For This Session.

If the Timestamp option is unselected when you click OK, a dialog box warns that the application will fail to install when the digital certificate expires. If you click Yes in response to the warning, time stamping is disabled. If you click No, the Timestamp option is automatically selected and time stamping is enabled.

For more information on creating a self-signed digital certificate, see Digitally signing an AIR file in the *Developing AIR Applications* guide.

You can also create an AIR Intermediate (AIRI) application without a digital signature. A user cannot install the application on a desktop, however, until you add a digital signature.

Prepare an AIRI package that will be signed later

- ❖ In the Digital Signature dialog box, select Prepare An AIRI Package That Will Be Signed Later, and click OK.

The digital signature status changes to indicate that you have chosen to prepare an AIRI package that will be signed later, and the Set button changes to a Change button.

If you choose to sign the application later, you will need to use the command-line AIR Developer Tool included with Flash and with the AIR SDK. For more information, see *Developing Adobe AIR 1.1 Applications with Adobe Flash CS4 Professional*.

HTML publishing templates

About HTML publishing templates

A Flash HTML template is a file that contains static HTML code and flexible template code consisting of a special type of variables (which differ from ActionScript variables). When you publish a SWF file, Flash replaces these variables with the values you select in the HTML tab of the Publish Settings dialog box and produces an HTML page with your SWF file embedded.

Flash includes templates, suitable for most users' needs, that eliminate the need to manually create an HTML page that displays the SWF file. For example, the Flash Only template is useful for testing your files in a browser. It places the SWF file on the HTML page so that you can view it through a web browser with the Flash Player installed.

To publish a new HTML page, use the same template and change the settings. You can create custom templates using any HTML editor. Creating a template is the same as creating a standard HTML page, except that you replace specific values pertaining to a SWF file with variables that begin with a dollar sign (\$).

Flash HTML templates have the following special characteristics:

- A one-line title that appears on the Template pop-up menu on the HTML tab of the Publish Settings dialog box.
- A longer description that appears when you click Info on the HTML tab of the Publish Settings dialog box.
- Template variables beginning with a dollar sign (\$) that specify where parameter values should be substituted when Flash generates the output file.

Note: Use a backslash and dollar sign (\\$) combination to use a dollar sign for another purpose in the document.

- HTML `object` and `embed` tags that follow the tag requirements of Microsoft Internet Explorer and Netscape® Communicator® or Navigator®, respectively. To display a SWF file properly on an HTML page, follow these tag requirements. Internet Explorer uses the `object` HTML tag to open a SWF file; Netscape uses the `embed` tag.

See also

[“Examples using object and embed tags”](#) on page 391

Customize HTML publishing templates

Modify HTML template variables to create an image map, a text report, or a URL report, or to insert custom values for some of the most common Flash HTML `object` and `embed` tag parameters (for browsers that use ActiveX controls and plug-ins, respectively).

Flash templates can include any HTML content for your application or even code for interpreters such as ColdFusion and ASP.

1 Using an HTML editor, open the Flash HTML template to change. These templates are in the following locations:

- Windows XP or Vista: *boot drive*:\Documents and Settings*user*\Local Settings\Application Data\Adobe\Flash CS4*language*\Configuration\HTML\. The Application Data folder is usually a hidden folder; you might need to change your Windows Explorer settings to see this folder.
- Mac OS X 10.3 and later: *Macintosh HD*/Applications/Adobe Flash CS4/*language*/First Run/HTML.

The *boot drive* is the drive from which Windows 2000 or Windows XP boots (usually C:). The *user* is the name of the person logged in to the Windows 2000 or Windows XP operating system. The *language* is set to an abbreviated language name. For example, in the US, *language* is set to “en” for English.

2 Edit the template.

3 Save the template in the same folder that you retrieved it from.

4 To apply the template settings to your SWF file, select File > Publish Settings, click HTML, and select the template you modified. Flash changes only the template variables in the template selected.

5 Select your remaining publish settings, and click OK.

See also

[“Publishing overview”](#) on page 379

HTML template variables

The following table lists the template variables that Flash recognizes:

Attribute/parameter	Template variable
Template title	\$TT
Template description start	\$DS
Template description finish	\$DF
Flash (SWF file) title	\$T1
Flash (SWF file) title for search engine metadata	\$TL
Description for search engine metadata	\$DC
Metadata XML string for use with search engines	\$MD
Width	\$WI
Height	\$HE
Movie	\$MO
HTML alignment	\$HA
Looping	\$LO
Parameters for object	\$PO
Parameters for embed	\$PE
Play	\$PL
Quality	\$QU
Scale	\$SC
Salign	\$SA
Wmode	\$WM
Devicefont	\$DE
Bgcolor	\$BG
Movie text (area to write movie text)	\$MT
Movie URL (location of SWF file URL)	\$MU
Image width (unspecified image type)	\$IW
Image height (unspecified image type)	\$IH
Image filename (unspecified image type)	\$IS
Image map name	\$IU
Image map tag location	\$IM
QuickTime width	\$QW
QuickTime height	\$QH
QuickTime filename	\$QN
GIF width	\$GW
GIF height	\$GH
GIF filename	\$GN

Attribute/parameter	Template variable
JPEG width	\$JW
JPEG height	\$JH
JPEG filename	\$JN
PNG width	\$PW
PNG height	\$PH
PNG filename	\$PN

Using shorthand template variables

The \$PO (for object tags) and \$PE (for embed tags) template variables are useful shorthand elements. Each variable causes Flash to insert into a template any nondefault values for some of the most common object and embed parameters, including PLAY (\$PL), QUALITY (\$QU), SCALE (\$SC), SALIGN (\$SA), WMODE (\$WM), DEVICEFONT (\$DE), and BGCOLOR (\$BG).

Sample HTML template

The following Default.HTML template file in Flash includes many of the commonly used template variables:

```
$TTFlash Only
$DS
Display Adobe SWF file in HTML.
$DF
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
$CS
<title>$TI</title>
</head>
<body bgcolor="$BG">
<!--url's used in the movie-->
$MU
<!--text used in the movie-->
$MT
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://fpdownload.adobe.com/pub/shockwave/cabs/flash/swflash.cab#version=7,0,0,0"
width="$WI" height="$HE" id="$TI" align="$HA">
<param name="allowScriptAccess" value="sameDomain" />
$PO
<embed $PEwidth="$WI" height="$HE" name="$TI" align="$HA" allowScriptAccess="sameDomain"
type="application/x-shockwave-flash" pluginspage="http://www.adobe.com/go/getflashplayer" />
</object>
</body>
</html>
```

See also

[“Specify publish settings for HTML documents”](#) on page 384

Create an image map to substitute for a SWF file

Flash can generate an image map to show any image and maintain the function of buttons that link to URLs. When an HTML template includes the `$IM` template variable, Flash inserts the image map code. The `$IU` variable identifies the name of the GIF, JPEG, or PNG file.

- 1 In your document, select the keyframe to use for the image map and label it **#Map** in the frame Property inspector (Window > Properties). Use any keyframe with buttons that have attached ActionScript 1.0 or 2.0 `getURL` actions.

If you don't create a frame label, Flash creates an image map using the buttons in the last frame of the SWF file. This option generates an embedded image map, not an embedded SWF file.

- 2 To select the frame to show the image map, do one of the following:
 - For PNG or GIF files, label the frame to appear as **#Static**.
 - For JPEG, during the publish operation, place the playhead on the frame to be used for display.
- 3 In an HTML editor, open the HTML template you'll modify.
- 4 Save your template.
- 5 Select File > Publish Settings, click Format, select a format for the image map, and click OK.

For example, insert the following code in a template:

```
$IM  
<img src=$IS usemap=$IU width=$IW height=$IH BORDER=0>
```

This might produce the following code in the HTML document that the Publish command creates:

```
<map name="mymovie">  
<area coords="130,116,214,182" href="http://www.adobe.com">  
</map>  

```

Creating text and URL reports

The `$MT` template variable causes Flash to insert all the text from the current SWF file as a comment in the HTML code. This is useful for indexing the content of a SWF file and making it visible to search engines.

The `$MU` template variable makes Flash generate a list of the URLs that actions in the current SWF file refer to and insert the list at the current location as a comment. This action lets link verification tools detect and verify the links in the SWF file.

Embedding search metadata

The `$TL` (SWF file title) and `$DC` (description metadata) template variables let you include search metadata in the HTML. This ability can make the SWF file more visible to search engines, and provide meaningful search results. Use the `$MD` template variable to include the search metadata as an XML string.

About exporting from Flash

Export SWF files

The Flash Export commands do not store export settings separately with each file, as does the Publish command. (To create all the files you need to put Flash content on the web, use the Publish command.)

Export Movie exports a Flash document to a still-image format, creates a numbered image file for every frame in the document, and exports the sound in a document to a WAV file (Windows only).

- 1 Open the Flash document to export, or select the frame or image to export in the current document.
- 2 Select File > Export > Export Movie, or File > Export > Export Image.
- 3 Enter a name for the output file.
- 4 Select the file format and click Save. If the format you selected requires more information, an Export dialog box appears.
- 5 Set the export options for the format you selected. See “[About export file formats](#)” on page 411.
- 6 Click OK, and then click Save.

See also

“[Publishing Flash documents](#)” on page 379

“[Creating multilanguage text](#)” on page 263

About export file formats

Remember the following:

- If the format you selected requires more information, an Export dialog box appears.
- When you save a Flash image as a bitmap GIF, JPEG, PICT (Macintosh), or BMP (Windows) file, the image loses its vector information and is saved with pixel information only. You can edit images exported as bitmaps in image editors such as Adobe® Photoshop®, but you can no longer edit them in vector-based drawing programs.
- When you export a Flash file in the SWF format, text is encoded as Unicode, providing support for international character sets, including double-byte fonts. Flash Player 6 and later versions support Unicode encoding.

Flash content is exported as sequences and images are exported as individual files. PNG is the only cross-platform bitmap format that supports transparency (as an alpha channel). Some non-bitmap export formats do not support alpha (transparency) effects or mask layers.

The following table lists the formats that you can export Flash content and images to:

File type	Extension	Windows	Macintosh
“ Adobe Illustrator Sequence and Illustrator Image ” on page 413	.ai	•	•
“ Animated GIF, GIF Sequence, and GIF Image ” on page 414	.gif	•	•
“ Bitmap (BMP) Sequence and Bitmap Image ” on page 414	.bmp	•	
“ DXF Sequence and AutoCAD DXF Image ” on page 414	.dxf	•	•
“ Enhanced Metafile (EMF) Sequence and Image (Windows) ” on page 414	.emf	•	
“ Encapsulated PostScript (EPS) 3.0 with Preview ” on page 415	.eps	•	•
“ Flash document (SWF) ” on page 415	.swf	•	•
“ JPEG Sequence and JPEG Image ” on page 412	.jpg	•	•
“ PICT Sequence and PICT Image (Macintosh) ” on page 413	.pct		•
“ PNG Sequence and PNG Image ” on page 413	.png	•	•
“ Export QuickTime ” on page 416	.mov	•	•

File type	Extension	Windows	Macintosh
“WAV audio (Windows)” on page 417	.wav	•	
“Windows AVI (Windows)” on page 417	.avi	•	
“Windows Metafile Image and Windows Metafile Sequence” on page 415	.wmf	•	

Update SWF files for Dreamweaver

To add the content to your page, export SWF files directly to an Adobe® Dreamweaver® site. Dreamweaver generates all the needed HTML code. You can start Flash from Dreamweaver to update the content. In Dreamweaver, you can update the Flash document (FLA file) and re-export the updated content automatically.

For more information on working with Dreamweaver, see *Using Dreamweaver* in Dreamweaver Help.

- 1 In Dreamweaver, open the HTML page that contains the Flash content.
- 2 Do one of the following:
 - Select the Flash content within the HTML page, and click Edit.
 - In Design view, press Control (Windows) or Command (Macintosh), and double-click the Flash content.
 - In Design view, right-click (Windows) or Control-click (Macintosh) the Flash content, and select Edit with Flash.
 - In Design view, in the Site panel, right-click (Windows) or Control-click (Macintosh) the Flash content, and select Open with Flash.
- 3 If the FLA file for the exported file does not open, the Open File dialog box appears. Navigate to the FLA file, and click Open.
- 4 If the user used the Change Link Sitewide feature in Dreamweaver, a warning appears. To apply link changes to the SWF file, click OK. To prevent the warning message from appearing when you update the SWF file, click Don't Warn Me Again.
- 5 Update the FLA file as needed in Flash.
- 6 To save the FLA file and reexport it to Dreamweaver, do one of the following:
 - To update the file and close Flash, click the Done button above the upper-left corner of the Stage.
 - To update the file and keep Flash open, select File > Update for Dreamweaver.

Exporting Images and Graphics

JPEG Sequence and JPEG Image

These options match the JPEG Publish Settings options with one exception: Match Screen matches the exported image to the size of the Flash content as it appears on your screen. Match Movie matches the JPEG image to the Flash content and maintains the aspect ratio of the original image.

See also

[“Specify publish settings for JPEG files”](#) on page 394

PICT Sequence and PICT Image (Macintosh)

The standard graphics format on the Macintosh; can contain bitmap or vector information. Use the Export PICT dialog box to set the following options:

Dimensions Sets the size of the exported bitmap image specified in pixels. The size you specify always has the same aspect ratio as your original image.

Resolution Sets the resolution in dpi and automatically calculates width and height based on the size of your drawing. To set the resolution to match your monitor, select Match Screen. Bitmap PICT images usually look best onscreen with 72-dpi resolution.

Include Export the minimum image area or the full document size.

Color Depth Designate whether the PICT file is object based or bitmap. Object-based images generally look better when printed, and scaling doesn't affect their appearance. Bitmap PICT images normally look best onscreen and can be manipulated in applications such as Adobe Photoshop. You can also select a variety of color depths with bitmap PICT files.

Include Postscript Available only for an object-based PICT file to include information that optimizes printing on a PostScript printer. This information makes the file larger and some applications might not recognize it.

Smooth Bitmap Available only for bitmap PICT images. Applies anti-aliasing to smooth jagged edges of a bitmap image.

PNG Sequence and PNG Image

The PNG export settings options are similar to the PNG Publish Settings options (which you can apply as well), with the following exceptions:

Dimensions Sets the size of the exported bitmap image to the number of pixels you enter in the Width and Height fields.

Resolution Enter a resolution in dpi. To use the screen resolution and maintain the aspect ratio of your original image, select Match Screen.

Colors The same as the Bit Depth option in the PNG Publish Settings tab and sets the number of bits per pixel to use in creating the image. For a 256-color image, select 8-bit; for thousands of colors, select 24-bpc; for thousands of colors with transparency (32 bpc) select 24-bpc with Alpha. The higher the bit depth, the larger the file.

Include Select to export the minimum image area or specify the full document size.

Filter Options match those in the PNG Publish Settings tab.

See also

[“Specify publish settings for PNG files”](#) on page 395

Adobe Illustrator Sequence and Illustrator Image

When you export a Flash image as a vector-graphic file (in Adobe® Illustrator® format), you preserve its vector information. You can edit these files in other vector-based drawing programs, but you can't import these images into most page-layout and word-processing programs.

Flash supports exporting Adobe Illustrator 88, 3, 5, and 6 formats. Flash does not support the Photoshop EPS format or EPS files that the Print command generates. Select the Adobe Illustrator version in the Export Adobe Illustrator dialog box.

Versions of the Adobe Illustrator format before version 5 do not support gradient fills, and only version 6 supports bitmaps.

See also

[“Import Adobe Illustrator files”](#) on page 74

Animated GIF, GIF Sequence, and GIF Image

The settings are the same as those on the GIF tab in the Publish Settings dialog box, with the following exceptions:

Resolution Set in dots per inch (dpi). To use the screen resolution, enter a resolution or click Match Screen.

Include Export the minimum image area or the full document size.

Colors Set the number of colors that can be used to create the exported image. The color choices are: black and white, 4, 6, 16, 32, 64, 128, or 256 colors; or Standard Color (the standard 216-color, web-safe palette).

Animation Available for the Animated GIF export format only. Enter the number of repetitions, where 0 repeats endlessly.

See also

[“Specify publish settings for Flash Player detection”](#) on page 392

Bitmap (BMP) Sequence and Bitmap Image

Create bitmap images for use in other applications. The Export Bitmap dialog box has the following options:

Dimensions Sets the size of the exported bitmap image in pixels. The size you specify always has the same aspect ratio as your original image.

Resolution Sets the resolution of the exported bitmap image in dots per inch (dpi) and automatically calculates width and height based on the size of your drawing. To set the resolution to match your monitor, select Match Screen.

Color Depth Specifies the bit depth of the image. Some Windows applications do not support the newer 32-bit per channel (bpc) depth for bitmap images; if you have problems using a 32-bpc format, use the 24-bpc format.

Smooth Applies anti-aliasing to the exported bitmap. Anti-aliasing produces a higher-quality bitmap image, but it can create a halo of gray pixels around an image placed on a colored background. Deselect if a halo appears.

DXF Sequence and AutoCAD DXF Image

Export as AutoCAD DXF 10 files, for additional editing in a DXF-compatible application.

This format has no definable export options.

Enhanced Metafile (EMF) Sequence and Image (Windows)

A graphics format available in Windows XP® and Windows Vista® that saves both vector and bitmap information. EMF supports the curves used in Flash drawings better than the older Windows Metafile format. However, some applications do not support this graphics format.

This format has no definable export options.

Windows Metafile Image and Windows Metafile Sequence

This standard Windows graphics format, which most Windows applications support, yields good results for importing and exporting files.

This format has no definable export options.

Encapsulated PostScript (EPS) 3.0 with Preview

You can export the current frame as an EPS 3.0 file for placement in another application, such as a page layout application. A PostScript printer can print an EPS file. You can include a bitmap preview with the exported EPS file for applications that can import and print the EPS files (such as Microsoft® Word® and Adobe® InDesign®) but that can't display them onscreen.

This format has no definable exporting options.

Flash document (SWF)

To place the Flash content in another application, such as Dreamweaver, export the entire document as a SWF file. Flash exports the SWF file using the current settings from the Flash tab of the Publish Settings for the FLA file.

See also

[“Publishing Flash documents”](#) on page 379

Exporting video and sound

Video for Adobe Flash Player (FLV)

Import or export a static video stream with encoded audio. Use with communications applications, such as video conferencing or files that contain screen-share encoded data exported from Adobe's Macromedia Flash Media Server.

When you export video clips with streaming audio in FLV format, the Streaming Audio settings compress the audio.

Files in the FLV format are compressed with the Sorensen codec.

See also

[“Specify publish settings for SWF files”](#) on page 383

[“The H.264, On2 VP6, and Sorensen Spark video codecs”](#) on page 285

Export a FLV file

- 1 Select the video clip in the Library panel.
- 2 Select Library > Properties and click Export.
- 3 Enter a name for the exported file. Select a location to save it to, click Save, and click OK.

About QuickTime

Flash provides two methods of exporting Flash documents as QuickTime:

QuickTime export Exports a QuickTime file that can be distributed as streaming video, on a DVD, or used in a video editing application such as Adobe® Premiere Pro®. QuickTime export is intended for users who want to distribute Flash content, such as animation, in the QuickTime video format.

Be aware that the performance of the computer used to export QuickTime video may affect the quality of the video. If Flash is unable to export each frame, it will drop frames, resulting in poor video quality. If you encounter dropped frames, try using a faster computer with more memory or reduce the frames per second of the Flash document.

Publish as QuickTime Creates an application with a Flash track in the same QuickTime format installed on your computer. This lets you combine the interactive features of Flash with the multimedia and video features of QuickTime in a single QuickTime 4 movie, which anyone with QuickTime 4 or later can view.

If you import a video clip (in any format) into a document as an embedded file, you can publish the document as a QuickTime movie. If you import a video clip in QuickTime format into a document as a linked file, you can also publish the document as a QuickTime movie.

You must set your Publish settings to Flash 5 or earlier to publish QuickTime with Flash track. This means you cannot use Flash features implemented after Flash 6.

Note: Beginning with QuickTime 7.1.3, Flash tracks were disabled by default. To publish a QuickTime file with a Flash track using QuickTime 7.1.3 or later, select Edit > Preferences > QuickTime Preferences > Advanced Media Types within QuickTime, and enable the use of Flash tracks.

Exports all layers in the Flash document as a single Flash track, unless the document contains an imported QuickTime movie. The imported QuickTime movie remains in QuickTime format in the exported application.

Export QuickTime

- 1 Select File > Export > Export Movie.
- 2 Specify settings for the QuickTime movie to export. By default, QuickTime export creates a movie file using the same dimensions as the source Flash document and exports the Flash document in its entirety. The Export QuickTime Video dialog box contains the following options:

Dimensions The width and height in pixels for the frames of the QuickTime movie. You can specify only the width or the height; the other dimension is automatically set to maintain the aspect ratio of your original document. To set both the width and the height independently of each other, deselect Maintain Aspect Ratio.

Note: If the dimensions of the video are particularly large (for example, 740 x 480 pixels), you may need to change the frame rate of the movie to avoid dropping frames.

Note: The Dimensions option you set in the QuickTime Export Settings dialog is for the width and height of the FLA file exported as video. The dimensions you set in the QuickTime Settings dialog specify the size of the exported QuickTime movie. If you do not change the size in the later dialog, it remains as “current” so you do not need to change it.

Ignore stage color Creates an alpha channel using the stage color. The alpha channel is encoded as a transparent track, letting you overlay the exported QuickTime movie on top of other content to alter the background color or scene.

To create a QuickTime video with an alpha channel, you must select a video Compression Type that supports 32-bit encoding with an alpha channel. Codecs that support this are Animation, PNG, Planar RGB, JPEG 2000, TIFF, or TGA. You must also select Million of Color+ from the Compressor/Depth setting. To set the compression type and color depth, click the Settings button in the Video category of the Movie Settings dialog box.

When last frame is reached Exports the entire Flash document as a movie file.

After time has elapsed The duration of the Flash document to export in hours:minutes:seconds:milliseconds.

QuickTime Settings Opens the advanced QuickTime settings dialog box. The Advanced settings let you specify custom QuickTime settings. In general, use the default QuickTime settings, as they provide optimal playback performance for most applications. To modify the QuickTime settings, see the documentation provided with Apple QuickTime Pro for information on the available video parameters.

3 Click export.

Windows AVI (Windows)

Exports a document as a Windows video but discards any interactivity. Good for opening a Flash animation in a video-editing application. Because AVI is a bitmap-based format, documents that contain long or high-resolution animations can quickly become very large.

The Export Windows AVI dialog box has the following options:

Dimensions Specifies a width and height, in pixels, for the frames of an AVI movie. Specify only the width or the height; the other dimension is automatically set to maintain the aspect ratio of your original document. To set both the width and the height, deselect Maintain Aspect Ratio.

Video Format Selects a color depth. Some applications do not yet support the Windows 32-bpc image format. If this format presents problems, use the older 24-bpc format.

Compress Video Select standard AVI compression options.

Smooth Applies anti-aliasing to the exported AVI movie. Anti-aliasing produces a higher-quality bitmap image, but it can cause a halo of gray pixels to appear around images when placed over a colored background. Deselect if a halo appears.

Sound Format Set the sample rate and size of the sound track, and whether to export in mono or stereo. The smaller the sample rate and size, the smaller the exported file, with a possible trade-off in sound quality.

See also

[“About compressing sounds for export”](#) on page 278

WAV audio (Windows)

Exports only the sound file of the current document to a single WAV file. You can specify the sound format of the new file.

To determine the sampling frequency, bit rate, and stereo or mono setting of the exported sound, select Sound Format. To exclude event sounds from the exported file, select Ignore Event Sounds.

Chapter 17: Best practices

Adobe® Flash® CS4 Professional contains features and capabilities that make it a flexible tool and allow more than one way to do the same thing in Flash. Over time, the Flash community has developed preferred methods for accomplishing many common tasks.

Best practices encourage consistency when you work on Flash or ActionScript® documents, share FLA or ActionScript files, work on applications, and when you are learning or teaching Flash and ActionScript.

Structuring FLA files

Organizing timelines and the library

Frames and layers on a timeline show you where assets are placed and determine how your document works. How a timeline and the library are set up and used affect the entire FLA file and its overall usability. The following guidelines help you author content efficiently, and let other authors who use your FLA documents have a greater understanding of how the document is structured.

- Give each layer an intuitive layer name, and place related assets together in the same location. Avoid using the default layer names (such as Layer 1, Layer 2).

Clearly describe the purpose or content of each layer or folder when you name them.

If applicable, place your layers that include ActionScript and a layer for frame labels at the top of the layer stack in the timeline. For example, name the layer that contains your ActionScript actions.

- Use layer folders to group and organize similar layers, to facilitate locating the layers that include code and labels.
- Lock layers that you are not using or do not want to modify. Lock your ActionScript layer immediately so that symbol instances or media assets are not placed on that layer.
- Never put any instances or assets on a layer that includes ActionScript. Because this can potentially cause conflicts between assets on the Stage and ActionScript that references them, keep all of your code on its own actions layer, and lock it after you create it.
- Use frame labels in a FLA file instead of using frame numbers in your ActionScript code if you reference frames in your code. If those frames change later when you edit the timeline, and you use frame labels and move them on the timeline, you do not have to change any references in your code.
- Use library folders.

Use folders in the library to organize similar elements (such as symbols and media assets) in a FLA file. If you name library folders consistently each time you create a file, it is easier to remember where you put assets. Commonly used folder names are Buttons, MovieClips, Graphics, Assets, Components, and, sometimes, Classes.

Using scenes

Using scenes is similar to using several SWF files to create a larger presentation. Each scene has a timeline. When the playhead reaches the final frame of a scene, the playhead progresses to the next scene. When you publish a SWF file, the timeline of each scene combines into a single timeline in the SWF file. After the SWF file compiles, it behaves as if you created the FLA file using one scene. Because of this behavior, avoid using scenes for the following reasons:

- Scenes can make documents confusing to edit, particularly in multiauthor environments. Anyone using the FLA document might have to search several scenes within a FLA file to locate code and assets. Consider loading content or using movie clips instead.
- Scenes often result in large SWF files.
- Scenes force users to progressively download the entire SWF file, instead of loading the assets they actually want to see or use. If you avoid scenes, the user can control what content they download as they progress through your SWF file. The user has more control over how much content they download, which is better for bandwidth management. One drawback is the requirement for managing a greater number of FLA documents.
- Scenes combined with ActionScript might produce unexpected results. Because each scene timeline is compressed onto a single timeline, you might encounter errors involving your ActionScript and scenes, which typically requires extra, complicated debugging.

If you create lengthy animations, you might find it advantageous to use scenes. If disadvantages apply to your document, consider using multiple FLA files, or movie clips to build an animation instead of using scenes.

Saving files and version control

When you save your FLA files, use a consistent naming scheme for your documents. This is particularly important if you save multiple versions of a single project.

Some problems might occur if you only work with one FLA file and do not save versions when you create the file. Files might become larger because of the history that's saved in the FLA file, or become corrupt (as with any software you use) while you are working on the file.

If you save multiple versions while developing, you have an earlier version available if you need to revert.

Use intuitive names for your files that are easy to read, not cryptic, and work well online:

- Do not use spaces, capitalization, or special characters.
- Only use letters, numbers, dashes, and underscores.
- If you save multiple versions of the same file, devise a consistent numbering system such as menu01.swf, menu02.swf and so on.
- Consider using all lowercase characters in your naming schemes, because some server software is case sensitive.
- Consider a naming system that uses a noun-verb or adjective-noun combination for naming files, for instance, classplanning.swf and myproject.swf.

Use the following methods to save new versions of a FLA file when you build an extensive project:

- Select File > Save As, and save a new version of your document.
- Use version control software or the Project panel to control your Flash documents.

You can use several options to save a file: Save, Save As, and Save and Compact. When you save a file, Flash does not analyze all the data before creating an optimized version of the document. Instead, the changes you make to the document are appended to the end of the FLA file's data, which shortens the time it takes to save the document.

When you Save As, Flash writes a new and optimized version of the file, which results in a smaller file size. When you Save and Compact, Flash creates a new optimized file (removing the undo history) and deletes the original file.

Important: *If you select Save when working with a document, you can undo before that save point. Because Save and Compact deletes the earlier version of the file and replaces it with the optimized version, you cannot undo earlier changes.*

If you are not using version control software to create backups of your FLA file, use Save As and type a new file name for your document after every milestone in your project.

Many software packages allow users to use version control with their files, which enables teams to work efficiently and reduce errors (such as overwriting files or working on old versions of a document). As with other documents, you can use these programs to organize the Flash documents outside Flash.

Organizing ActionScript in an application

Keeping actions together

Whenever possible, put your ActionScript® in a single location. Organizing your code in one place helps you edit projects more efficiently, because you can avoid searching in different places when you debug or modify the ActionScript. If you put code in a FLA file, put ActionScript on Frame 1 or Frame 2 in a layer called *actions* on the topmost layer in the Timeline. Alternatively, you might put all of your code in ActionScript files. Some Flash applications do not always put all code in a single place (in particular, ActionScript 2.0-based applications that use screens or behaviors).

You can usually put all your code in the same location (on a frame, or in ActionScript files), with the following advantages:

- Code is easy to find in a potentially complex source file.
- Code is easy to debug.

See also

[“Attaching code to objects”](#) on page 420

[“Behaviors conventions”](#) on page 421

[“Using the MVC design pattern”](#) on page 428

[“Organizing files and storing code”](#) on page 428

Attaching code to objects

Avoid attaching ActionScript to objects in a FLA file, even in simple SWF files. (Only ActionScript 1.0 and 2.0. can be attached to objects; ActionScript 3.0 cannot.) Attaching code to an object means that you select a movie clip, component, or button instance; open the Actions panel; and add ActionScript using the `on()` or `onClipEvent()` handler functions.

Attaching ActionScript code to objects is strongly discouraged for the following reasons:

- It is difficult to locate, and the FLA files are difficult to edit.
- It is difficult to debug.
- ActionScript that is written on the timeline or in classes is more elegant and easier to build upon.
- It encourages poor coding style.

- The contrast between two styles of coding can be confusing to people learning ActionScript; it forces students and readers to learn different coding styles, additional syntax, and a poor and limited coding style.

Avoid attaching ActionScript 2.0 to a button called `myButton_btn`, which looks like the following:

```
on (release) {  
    //do something  
}
```

However, placing ActionScript 2.0 with the same purpose on the timeline (which is encouraged), looks like the following code:

```
myButton_btn.onRelease = function() {  
    //do something  
};
```

Note: Different practices apply when using behaviors, which sometimes involves attaching code to objects.

See also

[“Comparing timeline code with object code”](#) on page 421

Behaviors conventions

About behaviors conventions

Behaviors are prewritten ActionScript 2.0 code snippets that you can add to parts of a FLA file. Many developers enter ActionScript code either into one or several frames on the main Timeline or in external ActionScript files. However, when you use behaviors, sometimes code is placed directly on symbol instances (such as buttons, movie clips, or components) instead of being placed on the timeline.

Behaviors are not supported by ActionScript 3.0.

Comparing timeline code with object code

To avoid problems that decentralized ActionScript 2.0 code creates, carefully plan a document that uses behaviors. Many developers do not place ActionScript on symbol instances, and instead place their code on the Timeline (timeline code) or in classes. Because behaviors add code to many locations in a FLA file, your ActionScript is not centralized and can be difficult to locate. When code is not centralized, it is difficult to understand interactions between the snippets of code, and it is impossible to write elegant code. Decentralized code can potentially lead to problems debugging code or editing files.

If you use behaviors, try the following features to facilitate working with behaviors and decentralized ActionScript:

Script Navigator Makes your timeline code or code on individual objects easy to find and edit in the Actions panel.

Find And Replace Lets you search for strings and replace them in a FLA file.

Script Pinning Lets you pin multiple scripts from various objects and work with them simultaneously in the Actions panel. This method works best with the Script navigator.

Movie Explorer Lets you view and organize the contents of a FLA file, and select elements (including scripts) for further modification.

When to use behaviors

The main difference between a FLA file with behaviors and a FLA file without behaviors is the workflow you must use for editing the project. If you use behaviors, you must select each instance on the Stage, or select the Stage, and open the Actions or Behaviors panel to make modifications. If you write your own ActionScript and put all your code on the main Timeline, you only have to make your changes on the Timeline.

If you have a FLA file with symbols, you can select one of the instances on the Stage, and use the Add menu on the Behaviors panel to add a behavior to that instance. The behavior you select automatically adds code that attaches to the instance, using “object code” such as the `on()` handler. You can also select a frame on a timeline, and add different behaviors to a frame using the Behaviors panel.

Decide how to structure your FLA file. Examine how and where to use behaviors and ActionScript in your FLA file. Consider the following questions:

- What code do the behaviors contain?
- Do you have to modify the behavior code? If so, by how much? To modify the behavior code to any extent, do not use behaviors. You usually cannot edit behaviors by using the Behaviors panel if you make modifications to the ActionScript. To significantly edit the behaviors in the Actions panel, it is usually easier to write all of the ActionScript yourself in a centralized location.
- What other ActionScript do you need, and does other ActionScript have to interact with the behavior code? Debugging and modifications are easier to make from a central location. For example, if code on a timeline interacts with behaviors placed on objects, avoid behaviors.
- How many behaviors do you have to use, and where do you plan to put them in the FLA file? If your behaviors are all placed on a timeline, they might work well in your document. Or, your workflow might not be affected if you use only a small number of behaviors. However, if you use many behaviors on a lot of object instances, writing your own code on the Timeline or in external ActionScript files might be more efficient.

Remember, ActionScript 3.0 does not support behaviors.

Using behaviors consistently

Use behaviors consistently throughout a document when they are your main or only source of ActionScript. Use behaviors when you have little or no additional code in the FLA file, or have a consistent system in place for managing the behaviors that you use.

If you add ActionScript to a FLA file, put code in the same locations where behaviors are added, and document how and where you add code.

For example, if you place code on instances on the Stage (object code), on the main Timeline (frame scripts), and also in external AS files, examine your file structure. Your project will be difficult to manage if you have code in all of these places. However, if you logically use behaviors and structure your code to work in a particular way surrounding those behaviors (place everything on object instances), at least your workflow is consistent. The document will be easier to modify later.

Sharing files that use behaviors

If you plan to share your FLA file with other users and you use ActionScript placed on or inside objects (such as movie clips), it can be difficult for those users to find your code’s location, even when they use the Movie Explorer to search through the document.

Document the use of behaviors if you are working with a complex document. Depending on the size of the application, create a flow chart, list, or use good documentation comments in a central location on the main Timeline.

If you are creating a FLA file with code placed in many locations throughout the document and plan to share the file, leave a comment on Frame 1 on the main Timeline to tell users where to find the code and how the file is structured. The following example shows a comment (on Frame 1) that tells users the location of the ActionScript:

```
/*  
    ActionScript placed on component instances and inside movie clips using behaviors.  
    Use the Movie Explorer to locate ActionScript  
*/
```

Note: *This technique is not necessary if your code is easy to find, the document is not shared, or all of your code is placed on frames of the main Timeline.*

Video conventions

About video conventions

Many options are available to edit video before you import it into a FLA document, or load an FLV file into a SWF file. Flash and Adobe Media Encoder have greater controls for video compression. Compressing video carefully is important because it controls the quality of the video footage and the size of the file. Video files, even when compressed, are large in comparison with most other assets in your SWF file.

Note: *Provide the user with control over the media in a SWF file. For example, if you add audio to a document with video (or even a looping background sound), let the user control the sound.*

Using video in an application

Before you import video into Flash, consider what video quality you need, what video format to use with the FLA file, and how to download it. When you import video into a FLA file (called *embedded video*), it increases the size of the SWF file that you publish. This video starts progressively downloading to the user's computer whether or not they view the video.

You can also progressively download or stream the video at runtime from an external FLV file on your server. When it starts downloading depends on how you structure your application.

Note: *Video progressively downloads from the server like SWF files, which is not actually streaming. Dynamically loading content has distinct advantages over keeping all your content in a single SWF file. For example, you will have smaller files and quicker loading, and the user only downloads what they want to see or use in your application.*

You can display external FLV video using a component or a video object. A component makes developing applications with FLV video easy, because the video controls are prebuilt, and you only need to specify an FLV file path to play the content. To keep your SWF file as small as possible, display video in a video object and create your own assets and code to control the video. Also consider using the FLVPlayback component in Adobe® Flash® CS4 Professional, which has a smaller file size than Media components (Flash MX Professional 2004 and later).

It is a good idea to give users some control (such as the ability to stop, pause, play, and resume the video, and control volume) over the video in a SWF file.

To gain certain kinds of flexibility over your video, such as manipulating the video with animation, or syncing various parts of it with the timeline, embed the video in the SWF file rather than loading it using ActionScript or one of the Media components.

For more control over a video instance than the Video class allows, place video inside a movie clip instance. The video's timeline plays independently from a Flash timeline, and you can place the content inside a movie clip to control timelines. You do not have to extend your main Timeline by many frames to accommodate for the video, which can make working with your FLA file difficult.

See also

[“Video”](#) on page 283

Exporting, compressing, and hosting FLV files

You can export FLV files from Adobe® Flash® CS4 Professional authoring environments. After you import video into your document, it appears as a video symbol in the library. To export the video as FLV, select the video and then select Library > Properties. To make export settings, click Export in the Embedded Video Properties dialog box.

The FLV QuickTime Exporter compresses video from third-party video editing software. You can also use Flash 8 Video Encoder, which is a stand-alone application that you can use to create FLV files. Because you have a lot of control over the export settings, the quality of the FLV file that is created using either of these tools is better than video exported directly from Flash.

When you compress video, remember the following recommendations:

- Do not recompress video.
Recompressing video leads to quality degradation, such as artifacts. Use raw footage or the least compressed footage that is available to you.
- Make your video as short as possible.
Trim the beginning and end of your video, and edit your video to remove any unnecessary content. This can be accomplished directly in Flash using the Video Import wizard.
- Adjust your compression settings.
If you compress footage and it looks great, try changing your settings to reduce the file size. Test your footage, and modify it until you find the best setting possible for the video you are compressing. All video has varying attributes that affect compression and file size; each video needs its own setting for the best results.
- Limit effects and rapid movement.
Limit movement if you are concerned about file size. Any kind of movement, particularly with many colors, increases file size. For example, effects (such as cross fades, blurs, and so on) increase file size, because the video contains more information.
- Choose appropriate dimensions.
If your target audience has a slow Internet connection (such as phone modems), make the dimensions of your video smaller, such as 160x120 pixels. If your visitors have fast connections, you can make your dimensions larger (for example, 320x240 pixels).
- Choose appropriate frames per second (fps).
If you target users that typically have older computer processors, choose a low rate of frames per second (such as 7 or 15 fps). If you target users with newer computers, you can use a higher rate of frames per second (such as 15 or 30 fps). Always choose an fps that is a multiple of your original frame rate. For example, if your original frame rate was 30 fps, compress to 15 fps or 7.5 fps.
- Choose an appropriate number of keyframes.

Video keyframes are different from keyframes in Flash. Each keyframe is a frame that draws when the video is compressed, so the more frequent your keyframes are the better quality the footage is. More keyframes also mean a larger file size. If you choose 30, a video keyframe draws every 30 frames. If you choose 15, the quality is higher because a keyframe draws every 15 frames and the pixels in your footage are more accurate to the original.

- Reduce noise.

Noise (scattered pixels in your footage) increases file size. Reduce noise using your video editor, to reduce the video file size. Using more solid colors in your video reduces its file size.

When you put FLV files on a server, they use the `video/x-flv` FLV mime type. If you have difficulty viewing FLV files after you upload your files, check that this mime type is set on your server. FLV files are binary, and some applications that you build might require that the `application/octet-stream` subtype is also set. For more information on the Flash Player specifications, see www.adobe.com/go/flashfileformat.

Troubleshooting video

You can create an application and then encounter problems after you upload it to your server.

- Check that your Flash Player version is correct.

For example, if you encoded your files using On2 codec, you need Flash Player 8 or later installed for the browsers you use to view your Flash content.

Note: For Flash Player and FLV compatibility, see [About using FLV video in Learning ActionScript 2.0 in Adobe Flash](#).

- Check that your server supports the FLV mime type. For more information on FLV files on a server, see [Configuring your server for FLV files in Learning ActionScript 2.0 in Adobe Flash](#).
- Check security guidelines.

If you load FLV files from another server, make sure that you have the proper files or code in place to load from that external server. For information on policy files, see [Server-side policy files for permitting access to data in Learning ActionScript 2.0 in Adobe Flash](#). For information about loading and security, see [Understanding security in Learning ActionScript 2.0 in Adobe Flash](#).

- Check your target paths to your video are correct. If you use relative paths (such as `/video/water.flv`), try using absolute paths (such as `http://www.helpexamples.com/flash/video/water.flv`). If your application doesn't work as a relative path, but does work as an absolute path, correct the relative path.

SWF application authoring guidelines

About SWF application guidelines

The best way to create Flash applications depends on the application you create and the technology that you are using to build the application.

An online application lets a user influence a website by interacting with it. For example, the application might collect information from the user (such as a username and password for a registration), information might be added to the site (such as in a forum), or the user might interact in real time with other site visitors (such as a chat room or interactive white board). Results from the server often appear in the SWF file, depending on the interaction. These examples are applications that involve the user and different kinds of server interaction. A website that does not use visitor information or data is not an application (for example, a portfolio, cartoon animation, or static informational site). Flash applications involve an interactive process between the user, a web application, and a server. The basic process is as follows:

- 1 A user enters information into a SWF file.
- 2 The information is converted into data.
- 3 The data is formatted and sent to a web server.
- 4 The data is collected by the web server and sent to an application server (for example, ColdFusion, PHP, or ASP).
- 5 The data is processed and sent back to the web server.
- 6 The web server sends the results to the SWF file.
- 7 The SWF file receives the formatted data.
- 8 Your ActionScript processes the data so the application can use it.


When you build an application, you must select a protocol for transferring data. The protocol alerts the application when data is sent or received, in what format the data is transferred, and how it handles a server's response. After data is received in the SWF file, it must be manipulated and formatted. If you use a protocol, you do not have to worry about data being in an unexpected format. When you transfer data using name-value pairs, you can check how the data is formatted. Check that the data is formatted correctly, so you do not receive XML formatted data and so the SWF file knows what data to expect and work with.

Collecting and formatting data

Applications depend on user interaction with the SWF file. Frequently, it depends on the user entering data into forms. Flash provides many ways you can enter and format data in Flash applications. This flexibility exists because of the capabilities you have with animation and creative control over the interface, and error checking and validation you can perform using ActionScript.

Benefits from using Flash to build forms to collect data include the following:

- Increased design control.
- Decreased or no need for page refreshing.
- Reuse of common assets.

 *To save information that you collect from the user, save it in a shared object on the user's computer. Shared objects let you store data on a user's computer, which is similar to using a cookie. For more information on Shared objects, see the `sharedObject` class in *ActionScript 2.0 Language Reference* or *ActionScript 3.0 Language and Components Reference*.*

Sending and processing data

You must typically process information before you send it to the server, so it's formatted in a way that the server understands. When the server receives the data, it can be manipulated in any number of ways and sent back to the SWF file in a format that it can accept, which can range from name-value pairs to complex objects.

Note: Your application server must have the MIME type of its output set to `application/x-www-urlform-encoded`. If that MIME type is missing, the result is usually unusable when it reaches Flash.

The following table shows you several options for sending data to a server and receiving data using Flash:

Send data	Description
<code>LoadVars.send</code> and <code>LoadVars.sendAndLoad</code>	Sends name-value pairs to a server-side script for processing. <code>LoadVars.send</code> sends variables to a remote script and ignores any response. <code>LoadVar.sendAndLoad</code> sends name-value pairs to a server and loads or parses the response into a target <code>LoadVars</code> object.
<code>XML.send</code> and <code>XML.sendAndLoad</code>	Similar to <code>LoadVars</code> , but <code>XML.send</code> and <code>XML.sendAndLoad</code> send XML packets instead of name-value pairs.
<code>getURL</code>	Using the <code>getURL()</code> function or <code>MovieClip.getURL</code> method, you can send variables from Flash to a frame or pop-up window.
Flash Remoting	Lets you easily exchange complex information between Flash and ColdFusion, ASP.NET, Java, and more. You can also use Flash Remoting to consume web services.
Web services	Adobe® Flash® CS4 Professional includes the <code>WebServiceConnector</code> component that lets you connect to remote web services, send and receive data, and bind results to components. This lets Flash developers quickly create Rich Internet Applications without having to write a single line of ActionScript. You can consume remote web services by using <code>WebServiceClasses</code> , which can require writing complex ActionScript.

Adding data loading and validation

Validate any information you retrieve before you send that data to a server. This reduces strain on the remote server, because it does not handle as many requests when users do not fill in required fields. Never rely solely on client-side validation in any application; server-side validation must also occur.

Even if you build a simple registration or login form, check that the user has entered their name and password. Perform this validation before sending the request to the remote server-side script and waiting for a result. Do not rely only on server-side validation. If a user enters only a username, the server-side script must receive the request, validate the data being sent, and return an error message to the Flash application, stating that it requires both the username and password. Likewise, if validation is performed only on the client side (within the SWF file), a user might hack the SWF file, bypass the validation, and send data to your server in an attempt to post the bad data.

Client-side validation can be as simple as making sure that a form field is at least one character long, or that the user entered a numeric value and not a string. To validate an e-mail address, for example, check that the text field in Flash isn't empty and contains at least the at sign (@) and dot (.) characters. For the server-side validation, add more complex validation and check that the e-mail address belongs to a valid domain.

You must write ActionScript to handle the data that loads into the SWF file from the server. After you finish loading data into a SWF file, the data can be accessed from that location. Use ActionScript to check whether the data is fully loaded. You can use callback functions or listeners to send a signal that the data is loaded into the document.

When you load data, it can be formatted in several ways:

- You might load XML, in which case you use the XML class methods and properties to parse the data and use it. If you use name-value pairs, the pairs turn into variables and you can manipulate them as variables.
- You might receive data from a web service or from Flash Remoting.

In both cases, you could receive complex data structures, such as arrays, objects, or record sets, which you must parse and bind appropriately.

Using error handling and debugging

Your application needs to be robust enough to anticipate certain errors and handle them accordingly.

One of the best ways to perform error handling in ActionScript 2.0 is to use the `try-catch-finally` blocks that let you throw and catch custom errors. By creating custom error classes, you can reuse code throughout your application without having to rewrite error handling code. For more information on throwing custom errors, see the `Error` class in *ActionScript*

2.0 Language Reference. For more information on `try-catch-finally` blocks, see `try...catch...finally` in *ActionScript 2.0 Language Reference*.

In ActionScript 3.0, use the `flash.errors` class to catch errors.

For more information, see “Handling synchronous errors in an application” in *Programming ActionScript 3.0*.

Organizing files and storing code

Consider the following guidelines before you start organizing files and storing code:

- Do you divide the SWF file into multiple SWF files, and, if so, how should they interact?
- What assets can you share across SWF files?
- What files do you dynamically load?
- How and where do you store ActionScript?

When you develop an application, store your server-side code and files in a logical directory structure, similar to those in an ActionScript package. Arrange your code this way to keep it well organized and reduce the risk of the code being overwritten.

For larger applications, encapsulate client-server communication and services in classes. When you use classes, you benefit in the following ways:

- You can reuse the code in more than one SWF file.
- You can edit code in a central place, and update all SWF files by republishing them.
- You can create a single API that can manipulate different UI elements or other assets that perform similar functions.

Using the MVC design pattern

The MVC design pattern is used to separate the information, output, and data processing in the application. The application is divided into three elements: model, view, and controller; each element handles a different part of the process.

The model Incorporates the data and rules of the application. Much of the application’s processing occurs in this part of the design pattern. The model also contains any components (such as CFCs, EJBs, and web services), and the database. Data returned is not formatted for the interface (or front end) of the application in this part of the process. The returned data can be used for different interfaces (or views).

The view Handles the front end of the application (the interface with which the user interacts), and renders the model’s contents. The interface specifies how the model’s data is presented and outputs the view for the user to use, and lets the user access or manipulate the application’s data. If the model changes, the view updates to reflect those changes by either pushing or pulling data (sending or requesting data). If you create a hybrid web application (for example, one that includes Flash interacting with other applications on the page), consider the multiple interfaces as part of the view in the design pattern. The MVC design pattern supports handling a variety of views.

The controller Handles the requirements of the model and view to process and display data, and typically contains a lot of code. It calls any part of the model, depending on user requests from the interface (or view), and contains code that's specific to the application. Because this code is specific to the application, it is usually not reusable. However, the other components in the design pattern are reusable. The controller does not process or output any data, but it takes the request from the user and decides what part of the model or view components it needs to call, and determines where to send the data and what formatting is applied to the returned data. The controller ensures that views have access to parts of the model data that they must display. The controller typically transmits and responds to changes that involve the model and view.

Each part of the model is built as a self-contained component in the overall process. If you change one part of the model (for example, you might rework the interface), the other parts of the process do not usually need modification, which reduces problems. If your design pattern is created correctly, you can change the view without reworking the model or controller. If your application does not use MVC, making changes anywhere can cause a rippling effect across all your code, which requires many more changes than if you were using a specific design pattern.

An important reason to use the MVC pattern is to separate data and logic from the user interface. By separating these parts of the process, you can have several different graphical interfaces that use the same model and unformatted data. This means that you can use your application with different Flash interfaces, such as an interface for the web, one for Pocket PC, a version for cell phones, and perhaps an HTML version that doesn't use Flash at all. Separating data from the rest of the application can greatly reduce the time it takes to develop, test, and even update more than one client interface. Similarly, adding new front ends for the same application is easier if you have an existing model to use.

Only use MVC if you build a large or complex application, such as an e-commerce website or an e-learning application. Using the architecture requires planning and understanding how Flash and this design pattern work. Carefully consider how the different pieces interact with each other; this typically involves testing and debugging. When you use MVC, testing and debugging are more involved and difficult than in typical Flash applications. If you build an application in which you need the additional complexity, consider using MVC to organize your work.

Creating secure applications

Dishonest users might try to hack your application, whether you build a small portal site where users can log in and read articles or a large e-commerce store. For this reason, consider the following steps to secure your application.

- Post data to HTTPS for data that needs to be secured. Encrypt values in Flash before sending them to a remote server to be processed.

Important: *Never store any information or code in a SWF file that you don't want users to see. It is easy to disassemble SWF files and view their contents using third-party software.*

- Add a cross-domain policy, which prevents unauthorized domains from accessing your assets.

Accessibility guidelines

About accessibility guidelines

Screen readers are complex, and you can easily encounter unexpected results in FLA files developed for use with screen readers, which is software that visually impaired users run to read websites aloud. Text is read aloud using specially designed software. A screen reader can only interpret textual content. However, any descriptions that you provide for the overall SWF file, movie clips, images, or other graphical content are also read aloud. Write descriptions for the important images and animations so that the screen reader can also interpret these assets in your SWF file. This is the SWF file equivalent to *alt* text in an HTML web page.

Note: Flash applications must be viewed in Internet Explorer on Windows, because Microsoft Active Accessibility (MSAA) support is limited to this browser.

Flash Player uses Microsoft Active Accessibility (MSAA) to expose Flash content to screen readers. MSAA is a Windows-based technology that provides a standardized platform for information exchange between assistive technologies, such as screen readers, and other applications. Events (such as a change in the application) and objects are visible to screen readers by using MSAA.

Note: Flash Player 7 (and later) does not work with all screen-reader technologies. The third-party software provider must handle the information that MSAA provides.

Creating accessible sites

Making a website accessible involves several different criteria:

Expose the information to screen readers **Make text or images realizable** Some visitors might have difficulty reading small text or seeing small graphics. Allow users to zoom in on these elements, taking advantage of scalable vector graphics in SWF files.

Provide audio narration Consider providing an audio narration for visitors without a screen reader, or where screen readers might not work, such as with video content.

Provide captions for audio narrations Some visitors might not be able to hear an audio narration for your site or a video. Consider providing captions for these visitors.

Do not rely on color to communicate information Many visitors might be color blind. If you rely on color to communicate information (such as: Click the green button to go to page 1, click the red button to go to page 2), provide text or speech equivalents.

Historically, many online presentations (such as videos) provide alternative ways for visually impaired visitors to access the content, for instance, a textual description of a video. However, Flash provides textual information directly to the screen reader. Although this usually means you need to make additional settings or ActionScript in a FLA file, you do not have to create a completely separate version.

Parts of your SWF file can be exposed to screen readers. Text elements (such as text fields, static text, and dynamic text), buttons, movie clips, components, and the entire SWF file can be interpreted by MSA-compliant screen readers.

Section 508 is United States legislation that provides guidelines for making information accessible to people with disabilities. Section 508 specifically addresses the need for websites to be accessible in several ways. Some websites, including all federal websites, must comply with these guidelines. If a SWF file does not communicate all of the information to the screen reader, the SWF file is no longer Section 508-compliant. For more information, see the Section 508 website.

Many nations have specified guidelines to follow to create accessible web sites, or follow guidelines established by other organizations. For more information on accessibility and web standards, see the World Wide Web Consortium (W3C) Web Accessibility Initiative website. These standards and guidelines describe what factors you must address when you create accessible HTML websites, and some of this information applies to Flash.

Exposing SWF file structure and navigation

Because of the visual nature of some SWF files, the layout and navigation of the page can be complex and difficult for screen readers to translate. An overall description of the SWF file is important to communicate information about its structure and how to navigate through the site's structure. You can provide this description by clicking the Stage and entering a description into the Accessibility panel. You can also create a separate area of the site to provide this description or overview.

Note: If you enter a description for the main SWF file, this description is read each time the SWF file refreshes. You can avoid this redundancy by creating a separate informational page.

Inform the user about any navigational elements that change in the SWF file. Perhaps an extra button is added, or the text on the face of a button changes, and this change is read aloud by the screen reader. Flash Player 7 and later supports updating these properties by using ActionScript. You can update the accessibility information in your applications if the content changes at runtime.

See also

[“Creating accessibility with ActionScript”](#) on page 312

Controlling descriptions and repetition

Designers and developers can assign descriptions for the animations, images, and graphics in a SWF file. Provide names for graphics so the screen reader can interpret them. If a graphic or animation does not communicate vital information to the SWF file (perhaps it is decorative or repetitive), or you outlined the element in the overall SWF file description, do not provide a separate description for that element. Providing unnecessary descriptions can be confusing to users who use screen readers.

Note: If you divide text or use images for text in your SWF files, provide either a name or description for these elements.

If you have several nested movie clips that serve a single purpose or convey one idea, ensure that you do the following:

- Group these elements in your SWF file.
- Provide a description for the parent movie clip.
- Make all the child movie clips inaccessible.

This is extremely important, or the screen reader tries to describe all the irrelevant nested movie clips, which will confuse the user, and might cause the user to leave your website. Make this decision whenever you have more than one object, such as many movie clips, in a SWF file. If the overall message is best conveyed using a single description, provide a description on one of the objects, and make all the other objects inaccessible to the screen reader.

Looping SWF files and applications cause screen readers to constantly refresh because the screen reader detects new content on the page. Because the reader thinks the content is updated, it returns to the top of the web page and starts rereading the content. Make inaccessible to screen readers any looping or refreshing objects that do not have to be reread.

Note: Do not type a description in the Description field of the Accessibility panel for instances (such as text) that the screen reader reads aloud.

See also

[“Using Flash to enter accessibility information for screen readers”](#) on page 305

Using color

You must make decisions about using colors in an accessible file. You must not rely only on color to communicate particular information or directives to users. A color-blind user cannot operate a page if it asks to click on the blue area to launch a new page or the red area to hear music. Offer text equivalents on the page or in an alternate version to make your site accessible. Also, check that significant contrast exists between foreground and background colors to enhance readability. If you place light gray text on a white background, users cannot easily read it. Similarly, small text is difficult for many visitors to read. Using high-contrast and large or resizable text benefits most users, even those without impairments.

Ordering, tabbing, and the keyboard

Reading order and tabbing are important considerations for making accessible Flash websites. When you design an interface, the order that it appears on the page might not match the order in which the screen reader describes each instance. You can control and test reading order, as well as control tabbing in the SWF file.

Controlling reading order

The default reading order is not predictable and does not always match the placement of your assets or the visual layout of the page. Keeping the layout simple can help create a logical reading order without using ActionScript. However, you have more control over reading order if you use ActionScript and test the reading order in your SWF files.

Important: Do not miss ordering a single instance in your SWF file, or the reading order reverts to the default (and unpredictable) reading order.

Controlling tabbing and content

Visitors who rely on screen readers to describe a site's content typically use tabbing and keyboard controls to navigate the operating system and web pages, because using the mouse is not useful when the screen cannot be seen. Use the `tabIndex` and `tabEnabled` properties with the movie clip, button, text field, or component instances to offer intelligent tabbing control in accessible SWF files. In addition to tabbing, you can use any key-press actions to navigate through the SWF file, but you must communicate that information using the Accessibility panel. Use the `Key` class in ActionScript to add key-press scripts to the SWF file. Select the object for which you want to use the key-press script, and add the shortcut key in the Shortcut field on the Accessibility panel. Add keyboard shortcuts to essential and frequently used buttons in your SWF file.

Note: In ActionScript 3.0, `tabIndex` and `tabEnabled` are properties of the `InteractiveObject` class. In ActionScript 2.0, they do not require a class reference.

Note: Avoid invisible buttons in accessible SWF files, because screen readers do not recognize these buttons. (Invisible buttons are buttons for which you define only a hit area, the clickable region, for the button.)

Many SWF files have a rapid succession of information, and screen readers frequently cannot keep up with this pace. Provide controls for the SWF file, letting the user use buttons to navigate through the file at their own pace, and letting them pause the process if necessary.

See also

[“Creating accessibility with ActionScript”](#) on page 312

Handling audio, video, and animation

When you provide audio narrations or video that contains speech, provide captions for those users who cannot hear. You can use text fields in Flash, import video that contains captions, or even use an XML caption file. You can use video cue points to specify when a text field should update text information at runtime.

For information on using Hi-Caption SE and the Hi-Caption Viewer component, see www.adobe.com/go/flash_extensions. This third-party extension lets you create captions that you save in an XML file and load into the SWF file at runtime, among other advanced controls. Alternatively, you can use cue points and a text field to display caption information.

See also

“[Accessibility for hearing-impaired users](#)” on page 304

Accessibility and extending Flash

With the extensibility layer in Flash, developers can create extensions that enable advanced authoring. This lets third-party companies develop extensions that involve accessibility. You have several options for validating your SWF files or adding captions.

For example, a validation tool can examine your SWF file for missing descriptions. It checks to see if a description is added for a group of instances, or if text has a label for the instance, and tells you about any problems. The tool also examines the reading order in your SWF file, and finds all instances that must be specified. You can specify reading order by using a dialog box after the SWF file is analyzed.

For information on the currently available third-party extensions, see www.adobe.com/go/flash_extensions.

Testing files and making changes

Test any SWF file that is intended for use with screen readers. Test your SWF files when each new version of Flash Player is released, including minor revisions, and test it with the following scenarios:

- Using the Window Eyes and JAWS for Windows screen readers. These each handle SWF files differently, so you could get different results.
- In a browser without a screen reader, and navigate through your site without using the mouse.
- Turn off your monitor and use only the screen reader to navigate your website.
- If you use audio narration, test your site without speakers.
- With several users who are representative of your target web site visitors.

Note: *You do not have to test different browsers, because the technology used to expose SWF files to screen readers (MSAA) is supported only by Internet Explorer on Windows.*

When listening to your SWF file using a screen reader, check the following points:

- Is the reading order correct?
- Do you have descriptions for shortcuts in your SWF file?
- Do you have adequate and complete descriptions for the elements in the interface?
- Do you have adequate descriptions for navigating the site’s structure?
- Is the SWF file content read when it is updated or refreshed?
- If you change the context of any elements on the Stage (such as a button that changes from Play to Pause), is that change announced by the screen reader?

No official tool is available for validating SWF files, unlike HTML validation. However, some third-party tools exist to help you validate the file. For more information on these extensions, see www.adobe.com/go/flash_extensions.

Advertising with Flash

Using recommended dimensions

Use the Interactive Advertising Bureau (IAB) guidelines to set dimensions for your Flash advertisements. The following table lists the recommended Interactive Marketing Unit (IMU) ad formats measurements:

Type of advertisement	Dimensions (pixels)
Wide skyscraper	160 x 600
Skyscraper	120 x 600
Half-page ad	300 x 600
Full banner	468 x 60
Half banner	234 x 60
Micro bar	88 x 31
Button 1	120 x 90
Button 2	120 x 60
Vertical banner	120 x 240
Square button	125 x 125
Leaderboard	728 x 90
Medium rectangle	300 x 250
Square pop-up	250 x 250
Vertical rectangle	240 x 400
Large rectangle	336 x 280
Rectangle	180 x 150

When you create a FLA file from a template (Select File > New, and click the Templates tab), you see many of these sizes.

Creating SWF file advertisements

Use these guidelines when you create advertisements:

- Optimize your graphics. Make SWF file banner advertisements 15K or smaller.
- Create a GIF banner advertisement in Flash that is 12K or smaller.
- Limit looping banner advertisements to three repetitions. Many websites adopt the standardized file size recommendations as advertising specifications.
- Use the `GET` command to pass data between an advertisement and a server, and do not use the `POST` command. For more information on GET and POST, see the `getURL` function in *ActionScript 2.0 Language Reference*.

Note: Provide control to the user. If you add sound to an advertisement, also add a mute button. If you create a transparent Flash ad that hovers over a web page, provide a button to close the advertisement for its duration.

See also

[“Optimizing graphics and animation”](#) on page 437

Tracking advertisements

Several leading advertising networks now support standardized tracking methods in Flash SWF files. The following guidelines describe the supported tracking methodology:

Create a button or movie clip button Use standardized dimensions outlined by the IAB. For a list of standardized dimensions, see the IAB website. For more information on creating a button in Flash, see [“Create a button”](#) on page 164.

Add a script to the button Executes when a user clicks the banner. You might use the `getURL()` function to open a new browser window. The following code snippets are two examples of ActionScript 2.0 code you might add to Frame 1 of the Timeline:

```
myButton_btn.onRelease = function() {  
    getURL(clickTAG, "_blank");  
};
```

You might add the following code to Frame 1 of the Timeline:

```
myButton_btn.onRelease = function() {  
    if (clickTAG.substr(0, 5) == "http:") {  
        getURL(clickTAG);  
    }  
};
```

The `getURL()` function adds the variable passed in the `object` and `embed` tags, and then sends the browser that is launched to the specified location. The server hosting the ad can track clicks on the advertisement. For more information on using the `getURL()` function, see *ActionScript 2.0 Language Reference*.

Assign clickTAG code for tracking Tracks the advertisement and helps the network serving the ad to track where the ad appears and when it is clicked.

The process is the standard way of creating an advertising campaign for a typical Flash advertisement. If you assign the `getURL()` function to the banner, you can use the following process to add tracking to the banner. The following example lets you append a variable to a URL string to pass data, which lets you set dynamic variables for each banner, instead of creating a separate banner for each domain. You can use a single banner for the entire campaign, and any server that is hosting the ad can track the clicks on the banner.

In the `object` and `embed` tags in your HTML, you would add code similar to the following example (where `www.helpexamples.com` is the ad network, and `adobe.com` is the company with an advertisement):

```
<EMBED src="your_ad.swf?clickTAG= http://helpexamples.com/tracking?http://www.adobe.com">
```

Add the following code in your HTML:

```
<PARAM NAME=movie VALUE="your_ad.swf?clickTAG =http://helpexamples.com/tracking?http://www.adobe.com">
```

For more information on advanced tracking techniques, see the Rich Media Advertising Center at www.adobe.com/go/rich_media_ads.

To download the Rich Media Tracking Kit, which includes examples and documentation, see www.adobe.com/go/richmedia_tracking.

To learn more about and download the Flash Ad Kit, which helps you deliver integrated and sophisticated advertisements, see www.adobe.com/go/learn_fl_flash_ad_kit.

Testing your ads

Test your SWF file ad on the most common browsers, especially the browsers that your target audience uses. Some users might not have Flash Player installed or they might have JavaScript disabled. Plan for these circumstances by having a replacement (default) GIF image or other scenarios for these users. For more information on detecting Flash Player, see “Specify publish settings for SWF files” on page 383. Give the user control of the SWF file. Let the user control any audio in the ad. If the advertisement is a borderless SWF file that hovers over a web page, let the user close the advertisement immediately and for the duration of the ad.

For the latest information on Flash Player version penetration for different regions, go to www.adobe.com/go/fp_version_penetration.

Optimizing FLA files for SWF output

Optimize Flash documents

As your document file size increases, so does its download time and playback speed. You can take several steps to prepare your document for optimal playback. As part of the publishing process, Flash automatically performs some optimization on documents. Before exporting a document, you can optimize it further by using various strategies to reduce the file size. You can also compress a SWF file as you publish it. As you make changes, test your document by running it on a variety of computers, operating systems, and Internet connections.

For a video tutorial about optimizing SWF files, see www.adobe.com/go/vid0140.

Optimize documents

- Use symbols, animated or otherwise, for every element that appears more than once.
- Use tweened animations whenever possible when creating animation sequences. Tweened animations use less file space than a series of keyframes.
- Use movie clips instead of graphic symbols for animation sequences.
- Limit the area of change in each keyframe; make the action take place in as small an area as possible.
- Avoid animating bitmap elements; use bitmap images as background or static elements.
- Use mp3, the smallest sound format, whenever possible.

Optimize elements and lines

- Group elements.
- Use layers to separate elements that change during the animation from elements that do not.
- Use Modify > Shape > Optimize to minimize the number of separate lines that are used to describe shapes.
- Limit the number of special line types, such as dashed, dotted, ragged, and so on. Solid lines require less memory. Lines created with the Pencil tool require less memory than brush strokes.

Optimize text and fonts

- Limit the number of fonts and font styles. Use embedded fonts sparingly because they increase file size.
- For Embed Fonts options, select only the characters needed instead of including the entire font.

Optimize colors

- Use the Color menu in the Symbol Property inspector to create many instances of a single symbol in different colors.
- Use the Color panel (Window > Color) to match the color palette of the document to a browser-specific palette.
- Use gradients sparingly. Filling an area with gradient color requires about 50 bytes more than filling it with solid color.
- Use alpha transparency sparingly because it can slow playback.

Speed up document display

To speed up the document display, use commands in the View menu to turn off rendering-quality features that require extra computing and slow down document display.

None of these commands have any effect on how Flash exports a document. To specify the display quality of Flash documents in a web browser, use the `object` and `embed` parameters. The Publish command can do this for you automatically.

❖ Select View > Preview Mode, and select from the following options:

Outlines Displays only the outlines of the shapes in your scene and causes all lines to appear as thin lines. This makes it easier to reshape your graphic elements and to display complex scenes quickly.

Fast Turns off anti-aliasing and displays all the colors and line styles of your drawing.

Antialias Turns on anti-aliasing for lines, shapes, and bitmaps and displays shapes and lines so that their edges appear smoother on the screen. Draws more slowly than the Fast option. Anti-aliasing works best on video cards that provide thousands (16-bit) or millions (24-bit) of colors. In 16- or 256-color mode, black lines are smoothed, but colors might look better in Fast mode.

Antialias Text Smooths the edges of any text. Works best with large font sizes and can be slow with large amounts of text. This is the most common mode in which to work.

Full Renders all content on the Stage fully. Might slow down display.

See also

[“Publishing overview”](#) on page 379

Optimizing graphics and animation

Before you create optimized and streamlined animations or graphics, outline and plan your project. Make a target for the file size and length of the animation, and test throughout the development process.

Follow these guidelines to optimize graphics and animation:

- Avoid using gradients, because they require many colors and calculations to be processed, which is more difficult for a computer processor to render.
- For the same reason, keep the amount of alpha or transparency you use in a SWF file to a minimum.

Animating objects that include transparency is processor-intensive and should be kept to a minimum. Animating transparent graphics over bitmaps is a particularly processor-intensive kind of animation, and must be kept to a minimum or avoided completely.

Note: The best bitmap format to import into Flash is PNG, which is the native file format of Macromedia Fireworks from Adobe. PNG files have RGB and alpha information for each pixel. If you import a Fireworks PNG file into Flash, you retain some ability to edit the graphic objects in the FLA file.

- Optimize bitmaps without overcompressing them. A 72-dpi resolution is optimal for the web. Compressing a bitmap image reduces file size, but compressing it too much compromises the quality of the graphic. Check that the settings for JPEG quality in the Publish Settings dialog box do not overcompress the image. Representing an image as a vector graphic is preferable in most cases. Using vector images reduces file size, because the images are made from calculations instead of many pixels. Limit the number of colors in your image while still retaining quality.

Note: Avoid scaling bitmaps larger than their original dimensions, because it reduces the quality of the image and is processor intensive.

- Set the `_visible` property to `false` instead of changing the `_alpha` level to 0 or 1 in a SWF file. Calculating the `_alpha` level for an instance on the Stage is processor intensive. If you disable the instance's visibility, it saves CPU cycles and memory, which can give your SWF files smoother animations. Instead of unloading and possibly reloading assets, set the `_visible` property to `false`, which is less processor-intensive.
- Reduce the number of lines and points you use in a SWF file. Use the Optimize Curves dialog box (Modify > Shape > Optimize) to reduce the number of vectors in a drawing. Select the Use Multiple Passes option for more optimization. Optimizing a graphic reduces file size, but compressing it too much compromises its quality. However, optimizing curves reduces your file size and improves SWF file performance. Third-party options are available for specialized optimization of curves and points that yield different results.

To get the best results, try different ways of producing animated content, and test each of the options.

A higher frame rate (measured in frames per second, or *fps*) produces smooth animation in a SWF file but it can be processor-intensive, particularly on older computers. Test your animations at different frame rates to find the lowest frame rate possible.

For a sample of scripted animation, see the Flash Samples web page at www.adobe.com/go/learn_fl_samples. Download and decompress the Samples zip file and navigate to the ActionScript2.0/Animation folder to access the sample.

See also

[“Animation frame rate and performance”](#) on page 438

[“Video conventions”](#) on page 423

Animation frame rate and performance

When you add animation to an application, consider the frame rate that you set your FLA file to. Frame rate can affect the performance of your SWF file and the computer that plays it. Setting a frame rate too high can lead to processor problems, especially when you use many assets or use ActionScript to create animation.

However, you also need to consider the frame rate setting, because it affects how smoothly your animation plays. For example, an animation set to 12 frames per second (fps) in the Property inspector plays 12 frames each second. If the document's frame rate is set to 24 fps, the animation appears to animate more smoothly than if it ran at 12 fps.

However, your animation at 24 fps also plays faster than it does at 12 fps, so the total duration (in seconds) is shorter. Therefore, to make a 5-second animation using a higher frame rate, you must add additional frames to fill those five seconds than at a lower frame rate (and thus, raises the total file size of your animation). A 5-second animation at 24 fps typically has a higher file size than a 5-second animation at 12 fps.

Note: When you use an `onEnterFrame` event handler to create scripted animations, the animation runs at the document's frame rate, similar to if you created a motion tween on a timeline. An alternative to the `onEnterFrame` event handler is `setInterval` (see *ActionScript 2.0 Language Reference*). Instead of depending on frame rate, you call functions at a specified interval. Like `onEnterFrame`, the more frequently you use `setInterval` to call a function, the more resource intensive the animation is on your processor.

Use the lowest possible frame rate that makes your animation appear to play smoothly at runtime, which helps reduce the strain on the end-user's processor. High frame rates (more than 30 to 40 fps) put a lot of stress on processors, and do not change the appearance of the animation much or at all at runtime.

Select a frame rate for your animation as early as possible in the development process. When you test the SWF file, check the duration, and the SWF file size, of your animation. The frame rate greatly affects the speed of the animation.

Filters and SWF file performance

If you use too many filters in an application, you can use large amounts of memory and cause Flash Player performance to suffer. Because a movie clip with filters attached has two bitmaps that are both 32-bit, these bitmaps can cause your application to use a significant amount of memory if you use many bitmaps. The computer's operating system might generate an out-of-memory error. On a modern computer, out-of-memory errors should be rare, unless you are using filter effects extensively in an application (for example, you have thousands of bitmaps on the Stage).

However, if you do encounter an out-of-memory error, the following occurs:

- The filters array is ignored.
- The movie clip is drawn using the regular vector renderer.
- No bitmaps are cached for the movie clip.

After an out-of-memory error occurs, a movie clip never attempts to use a filters array or a bitmap cache. Another factor that affects player performance is the value that you use for the quality parameter for each filter that you apply. Higher values require more CPU and memory for the effect to render, whereas setting the quality parameter to a lower value requires fewer computer resources. Avoid using an excessive number of filters, and use a lower quality setting when possible.

Important: If a 100 pixel by 100 pixel object is zoomed in once, it uses four times the memory since the content's dimensions are now 200 pixels by 200 pixels. If you zoom another two times, the shape is drawn as an 800 pixel by 800 pixel object which uses 64 times the memory as the original 100 pixel by 100 pixel object. Whenever you use filters in a SWF file, disable the zoom menu options from the SWF file's context menu.

You can encounter errors if you use invalid parameter types. Some filter parameters also have a particular valid range. If you set a value that's outside of the valid range, the value changes to a valid value that's within the range. For example, quality should be a value from 1 to 3 for a standard operation, and can only be set to 0 to 15. Anything higher than 15 is set to 15.

Some constructors have restrictions on the length of arrays required as input parameters. If a convolution filter or color matrix filter is created with an invalid array (not the right size), the constructor fails and the filter is not created successfully. If the filter object is then used as an entry on a movie clip's filters array, it is ignored.



When using a blur filter, using values for `blurX` and `blurY` that are powers of 2 (such as 2, 4, 8, 16, and 32) can be computed faster and give a 20% to 30% performance improvement.

Bitmap caching and SWF file performance

Bitmap caching helps you enhance the performance of nonchanging movie clips in your applications. When you set the `MovieClip.cacheAsBitmap` or `Button.cacheAsBitmap` property to `true`, Flash Player caches an internal bitmap representation of the movie clip or button instance. This can improve performance for movie clips that contain complex vector content. All of the vector data for a movie clip that has a cached bitmap is drawn to the bitmap, instead of to the main Stage.

***Note:** The bitmap is copied to the main Stage as unstretched, unrotated pixels snapped to the nearest pixel boundaries. Pixels are mapped one-to-one with the parent object. If the bounds of the bitmap change, the bitmap is re-created instead of being stretched.*

For detailed information on caching button or movie clip instances see the following topics:

- About caching and scrolling movie clips with ActionScript in [Learning ActionScript 2.0 in Adobe Flash](#)
- Caching a movie clip in [Learning ActionScript 2.0 in Adobe Flash](#)

Use the `cacheAsBitmap` property with movie clips with mostly static content and that do not scale and rotate frequently. With such movie clips, using the `cacheAsBitmap` property can lead to performance improvements when the movie clip is translated (when its `x` and `y` position is changed).

Enabling caching for a movie clip creates a *surface*, which has several advantages, such as helping complex vector animations to render fast. In some situations, enabling caching does not improve performance, or even decrease it.

Overall performance of cached data depends on how complex the vector data of your instances are, how much of the data you change, and whether or not you set the `opaqueBackground` property. If you are changing small regions, the difference between using a surface and using vector data might be negligible. Test both scenarios with your work before you deploy the application.

When to use bitmap caching

The following are typical scenarios in which you might see significant benefits when you enable bitmap caching by optimizing vector graphics.

Complex background image An application that contains a detailed and complex background image of vector data. To improve performance, select the content, store it in a movie clip, and set the `opaqueBackground` property to `true`. The background is rendered as a bitmap and can be redrawn quickly, so that your animation plays faster.

Scrolling text field An application that displays a large amount of text in a scrolling text field. Place the text field in a movie clip that you set as scrollable with scrolling bounds (the `scrollRect` property), enabling fast pixel scrolling for the specified instance. When a user scrolls the movie clip instance, the scrolled pixels shift up and generate the newly exposed region instead of regenerating the entire text field.

Windowing system An application with a complex system of overlapping windows. Each window can be open or closed (for example, web browser windows). If you mark each window as a surface (set the `cacheAsBitmap` property to `true`), each window is isolated and cached. Users can drag the windows so that they overlap each other, and each window doesn't need to regenerate the vector content.

When to avoid using bitmap caching

Misusing bitmap caching can negatively affect your SWF file. When you develop a FLA file that uses surfaces, remember the following guidelines:

- Do not overuse surfaces (movie clips with caching enabled). Each surface uses more memory than a regular movie clip; only enable surfaces to improve rendering performance.

- A cached bitmap can use significantly more memory than a regular movie clip instance. For example, if the movie clip on the Stage is 250 pixels by 250 pixels, when cached it might use 250 KB instead of 1 KB when it's a regular (uncached) movie clip instance.
- Avoid zooming in on cached surfaces. If you overuse bitmap caching, a large amount of memory is consumed (see previous bullet), especially if you zoom in on the content.
- Use surfaces for movie clip instances that are largely static (nonanimating). You can drag or move the instance, but the contents of the instance should not animate or change a lot. For example, if you rotate or transform an instance, the instance changes between the surface and vector data, which is difficult to process and negatively affects your SWF file.
- If you mix surfaces with vector data, it increases the amount of processing that Flash Player (and sometimes the computer) needs to do. Group surfaces together; for example, when you create windowing applications.

Working with components in Flash Player

The component framework lets you add functionality to components, but it can potentially add considerable file size to an application. Components inherit from each other. One component adds size to your Flash document, but subsequent components that use the same framework do not necessarily add more size. As you add components to the Stage, the file size increases, but at some point, it levels off because components share classes and do not load new copies of those classes.

If you use multiple components that do not share the same framework, they might add substantial file size to the SWF file. For example, the XMLConnector component adds 17K to the SWF file, and TextInput components add 24K to your document. If you add the ComboBox component, it adds 28K, because it is not part of the framework of either previous component. Because the XMLConnector component uses data binding, the classes add 6K to the SWF file. A document that uses all these components has 77K before you add anything else to the file. Carefully consider your SWF file size when you add a new component to the document.

Components must exist in the parent SWF file's library. For example, an application must have a copy of the components it uses in its library, even if those components are required only by child SWF files that are loaded at runtime. This is necessary to ensure that the components function properly, and slightly increases the download time of the parent SWF file. However, the parent library isn't inherited or shared in the SWF files that you load into the parent. Each child SWF file must download to the application with its own copy of the same components.

When you are planning to publish a SWF file with backward compatibility, you must have a good understanding of which components have that capability. The following table provides information about component availability in different versions of Flash Player:

Components	Flash Player 6 (6.0.65.0) and earlier	Flash Player 6 (6.0.65.0)	Flash Player 7 and 8	Flash Player 9
ActionScript 3.0	Not supported	Not supported	Not supported	Supported
ActionScript 2.0	Supported	Supported	Supported	Supported
V2 UI component set	Not supported	Supported	Supported	Supported
Media components	Not supported	Not supported	Supported	Supported
Data components	Not supported	Not supported	Supported	Supported

Deselect the Optimize for Flash Player 6r65 option in Publish Settings for the V2 UI components to work.

Optimizing component styles and performance

When using ActionScript 2.0, one of the most processor-intensive calls in a component framework is the `setStyle` call. The `setStyle` call executes efficiently, but the call is intensive because of the way it is implemented. The `setStyle` call is not always necessary in all applications, but if you use it, consider its performance effect.

To enhance performance, you can change styles before they are loaded, calculated, and applied to the objects in your SWF file. If you can change styles before the styles are loaded and calculated, you do not have to call `setStyle`.

To improve performance when using styles, set properties on each object as objects are instantiated. When you dynamically attach instances to the Stage, set properties in `initObj` in the call that you make to `createClassObject()`, as the following ActionScript shows:

```
createClassObject(ComponentClass, "myInstance", 0, {styleName:"myStyle", color:0x99CCFF});
```

For instances that you place directly on the Stage, you can use `onClipEvent()` for each instance, or you can use subclasses (recommended). For information on subclasses, see About writing a subclass in [Learning ActionScript 2.0 in Adobe Flash](#).

If you must restyle your components, you can improve efficiency in your application by using the Loader component. To implement several styles in different components, place each component in its own SWF file. If you change styles on the Loader component and reload the SWF file, the components in the SWF file are recreated. When the component is recreated, the cache of styles is emptied, and the style for the component is reset and referenced again.

Note: To apply a single style to all instances of a component in your SWF file, change the style globally using `_global.styles.ComponentName`.

Using runtime shared libraries

You can sometimes improve download time by using runtime shared libraries. These libraries are usually necessary for larger applications or when numerous applications on a site use the same components or symbols. By externalizing the common assets of your SWF files, you do not download classes repeatedly. The first SWF file that uses a shared library has a longer download time, because both the SWF file and the library load. The library caches on the user's computer, and then all the subsequent SWF files use the library. This process can greatly improve download time for some larger applications.

Displaying special characters

Computer operating systems have a specific code page that is regional. For example, a computer in Japan has a different code page than a computer in England. Flash Player 5 and earlier versions relied on the code page to display text; Flash Player 6 and later versions use Unicode to display text. Unicode is more reliable and standardized for displaying text because it is a universal character set that contains characters for all languages. Most current applications use Unicode.

You can use Unicode escape sequences to display special characters in Flash Player 6 and later. However, not all your characters display correctly if you do not load text that is UTF-8 or UTF-16 encoded (Unicode) or if you do not use a Unicode escape sequence to display the special character. For a set of Unicode code charts, see the Unicode web site at Unicode.org. For a list of commonly used escape sequences, see the table that follows in this section.

A non-Unicode application uses the operating system's code page to render characters on a page. In this case, the code page specifies the characters you see, so the characters appear correctly only when the code page on the user's operating system matches the application's code page. The code page that was used to create the SWF file needs to match the code page on the end user's computer. Using code pages is not a good idea for applications that an international audience might use; in this case, use Unicode instead.

Using `System.useCodepage` in your code forces the SWF file to use the system’s code page instead of Unicode.

Only use this process when you are loading non-Unicode encoded text from an external location and when this text is encoded with the same code page as the user’s computer. If both these conditions are true, the text appears without a problem. If both of these conditions are not true, use Unicode and a Unicode escape sequence to format your text. To use an escape sequence, add the following ActionScript 2.0 on Frame 1 of the Timeline:

```
this.createTextField("myText_txt", 99, 10, 10, 200, 25);
myText_txt.text = "this is my text, \u00A9 2004";
```

This ActionScript creates a text field, and enters text that includes a copyright symbol (©) into the text field.

You can make a SWF file use the operating system’s code page, which is controlled by the `useCodepage` property. When Flash exports a SWF file, it defaults to exporting Unicode text and `System.useCodepage` is set to `false`. You might encounter problems displaying special text, or text on international systems, where using the system’s code page can seem to solve the problem of text incorrectly displaying. However, using `System.useCodePage` is always a last resort.

To use the system’s code page, place the following line of ActionScript 2.0 code on Frame 1 of the Timeline:

```
System.useCodepage = true;
```

Important: A special character can appear only if the user’s computer has the character included in the font that is being used. If you are not sure, embed the character or font in the SWF file.

The following table contains a number of commonly used Unicode escape sequences.

Character description	Unicode escape sequence
em-dash (–)	\u2014
registered sign (®)	\u00AE
copyright sign (©)	\u00A9
trademark sign (™)	\u2122
Euro sign (€)	\u20AC
backslash (\)	\u005C
forward slash (/)	\u002F
open curly brace ({)	\u007B
close curly brace (})	\u007D
greater than (>)	\u003E
less than (<)	\u003C
asterisk (*)	\u002A

Test document download performance

Flash Player attempts to meet the frame rate you set; the actual frame rate during playback can vary on different computers. If a document that is downloading reaches a particular frame before the frame’s required data is downloaded, the document pauses until the data arrives.

To view downloading performance graphically, use the Bandwidth Profiler, which shows how much data is sent for each frame according to the modem speed you specify.

In simulating the downloading speed, Flash uses estimates of typical Internet performance, not the exact modem speed. For example, if you select to simulate a modem speed of 28.8 Kbps, Flash sets the actual rate to 2.3 Kbps to reflect typical Internet performance. The profiler also compensates for the added compression support for SWF files, which reduces the file size and improves streaming performance.

When external SWF files, GIF and XML files, and variables are streamed into a player by using ActionScript calls such as `loadMovie` and `getUrl`, the data flows at the rate set for streaming. The stream rate for the main SWF file is reduced based on the reduction of bandwidth that the additional data requests cause. Test your document at each speed and on each computer that you plan to support to ensure that the document doesn't overburden the slowest connection and computer for which it is designed.

You can also generate a report of frames that are slowing playback and then optimize or eliminate some of the content in those frames.

To change the settings for the SWF file created using the Test Movie and Test Scene commands, use File > Publish Settings.

See also

[“Optimize Flash documents”](#) on page 436

[“Publishing overview”](#) on page 379

[“Debugging ActionScript 1.0 and 2.0”](#) on page 350

[“Debugging ActionScript 3.0”](#) on page 360

Test download performance

1 Do one of the following:

- Select Control > Test Scene or Control > Test Movie.

If you test a scene or document, Flash publishes the current selection as a SWF file using the settings in the Publish Settings dialog box. The SWF file opens in a new window and begins playing immediately.

- Select File > Open, and select a SWF file.

2 Select View > Download Settings, and select a download speed to determine the streaming rate that Flash simulates. To enter a custom user setting, select Customize.

3 When viewing the SWF file, select View > Bandwidth Profiler to show a graph of the downloading performance.

The left side of the profiler displays information about the document, its settings, its state, and streams, if any are included in the document.

The right section of the profiler shows the Timeline header and graph. In the graph, each bar represents an individual frame of the document. The size of the bar corresponds to that frame's size in bytes. The red line beneath the Timeline header indicates whether a given frame streams in real time with the current modem speed set in the Control menu. If a bar extends above the red line, the document must wait for that frame to load.

4 Select View > Simulate Download to turn streaming off or on.

If you turn streaming off, the document starts over without simulating a web connection.

5 Click a bar on the graph to show settings for the corresponding frame in the left window and stop the document.

6 If necessary, adjust the view of the graph by taking one of the following actions:

- Select View > Streaming Graph to show which frames cause pauses.

This default view displays alternating light and dark gray blocks that represent each frame. The side of each block indicates its relative byte size. The first frame stores a symbol's contents, so it is often larger than other frames.

- Select View > Frame by Frame Graph to display the size of each frame.

This view helps you see which frames contribute to streaming delays. If any frame block extends above the red line in the graph, Flash Player stops playback until the entire frame downloads.

7 Close the test window to return to the authoring environment.

After you set up a test environment using the Bandwidth Profiler, you can open any SWF file directly in the test environment. The file opens in a Flash Player window, using the Bandwidth Profiler and other selected viewing options.

Generate a final report

- 1 Select File > Publish Settings, and click the Flash tab.
- 2 Select Generate Size Report.
- 3 Click Publish.

Flash generates a text file with the .txt extension. (If the document file is myMovie.fla, the text file is myMovie Report.txt.) The report lists the size of each frame, shape, text, sound, video and ActionScript script by frame.

Tips for creating content for mobile devices

Creating Flash content for use on mobile devices

To create Flash content for mobile devices, follow some basic principles. For example, Flash developers often avoid extremely complex artwork and excessive tweening or transparency.

Flash Lite developers face additional challenges because performance on mobile devices varies greatly. If content must be published to many different devices, developers sometimes have to author for the lowest common denominator.

Optimizing mobile content requires making trade-offs. For example, one technique may make the content look better, while another results in better performance. As you evaluate these trade-offs, you will be going back and forth repeatedly between testing in the emulator and testing on the target device. You must see your content on the actual device to evaluate the trueness of colors, text readability, physical interactions, UI responsiveness, and other aspects of the real mobile experience.

For more tips and techniques for creating content for mobile phones and devices, see www.adobe.com/go/learn_cs_mobilewiki_en.

Flash Lite guidelines for animation in mobile devices

When creating animated content for a mobile devices, keep device CPU limitations in mind. Following these guidelines can help prevent Flash Lite content from running slowly:

- When creating a new Flash Lite file, check that the document is set up correctly. Although Flash files scale smoothly, performance can suffer if the file is not running at its native Stage size and has to scale in the player. Try to set the document Stage size to match the resolution of the target device. Also set the Flash Player to the correct version of Flash Lite and select an appropriate device profile in Device Central.
- Flash Lite can render vector graphics at low, medium, and high quality. The higher the rendering quality, the more smoothly and accurately Flash Lite renders vector graphics and the greater the demand on the device's CPU. To provide complex animation, experiment with changing the player's quality setting and then thoroughly test the SWF file. To control the rendering quality of a SWF file, use the `_quality` property or the `SetQuality` command. Valid values for the `_quality` property are `LOW`, `MEDIUM`, and `HIGH`.
- Limit the number of simultaneous tweens. Reduce the number of tweens, or sequence the animation so that one begins when another ends.
- Use transparency (alpha) effects on symbols sparingly because they are CPU intensive. In particular, avoid tweening symbols with alpha levels that are not fully opaque (less than 100%).
- Avoid CPU-intensive visual effects, such as large masks, extensive motion, alpha blending, extensive gradients, and complex vectors.
- Experiment with combinations of tweens, keyframe animations, and ActionScript-driven movement to produce the most efficient results.
- Rendering vector ovals and circles is much more memory intensive than rendering quadrangles. Using round and oval strokes also greatly increases CPU use.
- Test animations frequently on actual target devices.
- When Flash draws an animated region, it defines a rectangular bounding box around the area. Optimize the drawing by making that rectangle as small as possible. Avoid overlapping tweens, because Flash interprets the merged area as a single rectangle, resulting in a larger total region. Use Flash's Show Redraw Region feature to optimize the animation.
- Avoid using `_alpha = 0` and `_visible = false` to hide on-screen movie clips. If you simply turn a movie clip's visibility off or change its alpha to zero, it is still included in line-rendering calculations, which can affect performance.
- Similarly, do not try to hide a movie clip by obscuring it behind another piece of artwork. It will still be included in the player's calculations. Instead, move movie clips completely off the Stage or remove them by calling `removeMovieClip`.

For more tips and techniques for creating content for mobile phones and devices, see www.adobe.com/go/learn_cs_mobilewiki_en.

Flash Lite bitmap and vector graphics in mobile devices

Flash Lite can render both vector and bitmap graphics. Each type of graphic has its advantages and disadvantages. The decision to use vector rather than bitmap graphics is not always clear and often depends on several factors.

Vector graphics are compactly represented in SWF files as mathematical equations and rendered at run time by the Flash Lite player. In contrast, bitmap graphics are represented as arrays of picture elements (pixels), which require more bytes of data. Therefore, using vector graphics in a file can help reduce file size and memory usage.

Vector graphics also maintain their smooth shapes when scaled in size. Bitmap images can appear boxy, or pixelated, when scaled.

Compared to bitmaps, vector graphics require more processing power to render, especially vector graphics that have many complex shapes and fills. Consequently, heavy use of vector shapes can sometimes reduce overall file performance. Because bitmap graphics do not require as much processing time to render as vector graphics, they are a better choice for some files, for example, a complex road map meant to be animated and scrolled on a mobile phone.

Keep these considerations in mind:

- Avoid using outlines on vector shapes. Outlines have an inner and outer edge (fills have only one) and are twice the work to render.
- Corners are simpler to render than curves. When possible, use flat edges, especially with very small vector shapes.
- Optimization is especially helpful with small vector shapes such as icons. Complex icons may lose their details upon rendering, and the work of rendering the details is wasted.
- As a general rule, use bitmaps for small, complex images (such as icons) and vector graphics for larger and simpler ones.
- Import bitmap graphics at the correct size; don't import large graphics and scale them down in Flash, because this wastes file size and run-time memory.
- The Flash Lite player does not support bitmap smoothing. If a bitmap is scaled or rotated, it will have a chunky appearance. If it is necessary to scale or rotate a graphic, consider using a vector graphic instead.
- Text is essentially just a very complex vector shape. Of course, text is often critical, so it can rarely be avoided entirely. When text is needed, avoid animating it or placing it over an animation. Consider using text as a bitmap. For multiline dynamic and input text, the line break of the text string is not cached. Flash breaks lines at run time and recalculates the breaks every time the text field needs to be redrawn. Static text fields are not problematic, because the line breaking is precalculated at compile time. For dynamic content, using dynamic text fields is unavoidable, but when possible, consider using static text fields instead.
- Minimize the use of transparency in PNG files; Flash must calculate redraws even for the transparent portions of the bitmap. For example, with a transparent PNG file that represents a foreground element, don't export the transparent PNG at the full size of the screen. Instead, export it at the actual size of the foreground element.
- Try to group bitmap layers together and vector layers together. Flash needs to implement different renderers for bitmap and vector content, and switching between renderers takes time.

For more tips and techniques for creating content for mobile phones and devices, see www.adobe.com/go/learn_cs_mobilewiki_en.

Set compression of Flash Lite bitmaps for mobile devices

When using bitmaps, you can set image-compression options (on a per-image basis or globally for all bitmap images) that reduce SWF file size.

For more tips and tricks about using Adobe Device Central with other Adobe products, see www.adobe.com/go/learn_cs_mobilewiki_en.

Set compression options for an individual bitmap file

- 1 Start Flash and create a document.
- 2 Select a bitmap in the Library window.
- 3 Right-click (Windows) or Control-click (Macintosh) the bitmap icon in the Library window, and select Properties from the context menu to open the Bitmap Properties dialog box.

4 In the Compression pop-up menu, select one of the following options:

- Select the Photo (JPEG) option for images with complex color or tonal variations, such as photographs or images with gradient fills. This option produces a JPEG file. Select the Use Imported JPEG Data check box to use the default compression quality specified for the imported image. To specify a new quality compression setting, deselect Use Imported JPEG Data and enter a value between 1 and 100 in the Quality text box. A higher setting produces an image of higher image, but also a larger file, so adjust the value accordingly.
- Select the Lossless (PNG/GIF) option for images with simple shapes and a few colors. This option compresses the image using lossless compression, which discards no data.

5 Click Test to determine the results of the file compression.

Compare the original file size to the compressed file size to decide whether the selected compression setting is acceptable.

Set compression for all bitmap images

- 1 Select File > Publish Settings, and then click the Flash tab to display compression options.
- 2 Adjust the JPEG quality slider, or enter a value. A higher JPEG quality value produces an image of higher image quality but a larger SWF file. A lower image quality produces a smaller SWF file. Try different settings to determine the best trade-off between size and quality.

Optimizing Flash Lite frames for mobile devices

- Most devices that support Flash Lite play back content at about 15 to 20 frames per second (fps). The frame rate can be as low as 6 fps. During development, set the document frame rate to approximate the playback speed of the target device. This shows how the content will run on a device with limited performance. Before publishing a final SWF file, set the document frame rate to at least 20 fps or higher to avoid limiting performance in case the device supports a higher frame rate.
- When using `gotoAndPlay`, remember that every frame between the current frame and the requested frame needs to be initialized before Flash plays the requested frame. If many of these frames contain different content, it could be more efficient to use different movie clips rather than using the Timeline.
- Although preloading all content by putting it at the beginning of the file makes sense on the desktop, preloading on a mobile device can delay file startup. Space content throughout the file so that movie clips are initialized as they are used.

For more tips and techniques for creating content for mobile phones and devices, see www.adobe.com/go/learn_cs_mobilewiki_en.

Optimizing ActionScript for Flash Lite content on mobile devices

Because of the processing speed and memory limitations on most mobile devices, follow these guidelines when developing ActionScript for Flash Lite content used on mobile devices:

- Keep the file and its code as simple as possible. Remove unused movie clips, delete unnecessary frame and code loops, and avoid too many frames or extraneous frames.
- Using FOR loops can be expensive because of the overhead incurred while the condition is checked with each iteration. When the costs of the iteration and the loop overhead are comparable, execute multiple operations individually instead of using a loop. The code may be longer, but performance will improve.
- Stop frame-based looping as soon as it is no longer needed.
- When possible, avoid string and array processing because it can be CPU-intensive.

- Always try to access properties directly rather than using ActionScript getter and setter methods, which have more overhead than other method calls.
- Manage events wisely. Keep event listener arrays compact by using conditions to check whether a listener exists (is not `null`) before calling it. Clear any active intervals by calling `clearInterval`, and remove any active listeners by calling `removeListener` before removing content using `unloadApplication` or `removeApplicationClip`. Flash does not re-collect SWF data memory (for example, from intervals and listeners) if any ActionScript functions are still referring to the SWF data when a movie clip is unloaded.
- When variables are no longer needed, delete them or set them to `null`, which marks them for garbage collection. Deleting variables helps optimize memory use during run time, because unneeded assets are removed from the SWF file. It is better to delete variables than to set them to `null`.
- Explicitly remove listeners from objects by calling `removeListener` before garbage collection.
- If a function is being called dynamically and passing a fixed set of parameters, use `call` instead of `apply`.
- Make namespaces (such as paths) more compact to reduce startup time. Every level in the package is compiled to an `IF` statement and causes a new `Object` call, so having fewer levels in the path saves time. For example, a path with the levels `com.xxx.yyy.aaa.bbb.ccc.functionName` causes an object to be instantiated for `com.xxx.yyy.aaa.bbb.ccc`. Some Flash developers use preprocessor software to reduce the path to a unique identifier, such as `58923409876.functionName`, before compiling the SWF code.
- If a file consists of multiple SWF files that use the same ActionScript classes, exclude those classes from select SWF files during compilation. This can help reduce file download time and run-time memory requirements.
- Avoid using `Object.watch` and `Object.unwatch`, because every change to an object property requires the player to determine whether a change notification must be sent.
- If ActionScript code that executes on a keyframe in the timeline requires more than 1 second to complete, consider splitting up that code to execute over multiple keyframes.
- Remove `trace` statements from the code when publishing the SWF file. To do this, select the Omit Trace Actions check box on the Flash tab in the Publish Settings dialog box.
- Inheritance increases the number of method calls and uses more memory: a class that includes all the functionality it needs is more efficient at run time than a class that inherits some of its functionality from a superclass. Therefore, you may need to make a design trade-off between extensibility of classes and efficiency of code.
- When one SWF file loads another SWF file that contains a custom ActionScript class (for example, `foo.bar.CustomClass`) and then unloads the SWF file, the class definition remains in memory. To save memory, explicitly delete any custom classes in unloaded SWF files. Use the `delete` statement and specify the fully qualified class name, such as: `delete foo.bar.CustomClass`.
- Limit the use of global variables, because they are not marked for garbage collection if the movie clip that defined them is removed.
- Avoid using the standard user interface components (available in the Components panel in Flash). These components are designed to run on desktop computers and are not optimized to run on mobile devices.
- Whenever possible, avoid deeply nested functions.
- Avoid referencing nonexistent variables, objects, or functions. Compared to the desktop version of Flash Player, Flash Lite 2 looks up references to nonexistent variables slowly, which can significantly affect performance.
- Avoid defining functions using anonymous syntax. For example, `myObj.eventName = function{ ...}`. Explicitly defined functions are more efficient, such as `function myFunc { ...}; myObj.eventName = myFunc;`

- Minimize the use of Math functions and floating-point numbers. Calculating these values slows performance. If you must use the Math routines, consider precalculating the values and storing them in an array of variables. Retrieving the values from a data table is much faster than having Flash calculate them at run time.

For more tips and techniques for creating content for mobile phones and devices, see www.adobe.com/go/learn_cs_mobilewiki_en.

Managing Flash Lite file memory for mobile devices

Flash Lite regularly clears from memory any objects and variables that a file no longer references. This is known as garbage collection. Flash Lite runs its garbage-collection process once every 60 seconds, or whenever usage of file memory increases suddenly by 20% or more.

Although you cannot control how and when Flash Lite performs garbage collection, you can still free unneeded memory deliberately. For timeline or global variables, use the `delete` statement to free the memory that ActionScript objects use. For local variables—for example, a variable defined within a function definition—you can't use the `delete` statement to free an object's memory, but you can set to `null` the variable that references the object. This frees the memory that the object uses, provided there are no other references to that object.

The following two code examples show how to free memory that objects use by deleting the variable that references those objects. The examples are identical, except that the first example creates a timeline variable and the second creates a global variable.

```
// First case: variable attached to a movie or
// movie clip timeline
//
// Create the Date object.
var mcDateObject = new Date();
// Returns the current date as a string.
trace(mcDateObject);
// Delete the object.
delete mcDateObject;
// Returns undefined.
trace(mcDateObject);
//
// Second case: global variable attached to a movie or
// movie clip timeline
//
// Create the Date object.
_global.gDateObject = new Date();
// Returns the current date as a string.
trace(_global.gDateObject);
// Delete the object.
delete _global.gDateObject;
// Returns undefined.
trace(_global.gDateObject);
```

As mentioned previously, you can't use the `delete` statement to free memory that a local function variable uses. Instead, set the variable reference to `null`, which has the same effect as using `delete`.

```
function func()
{
    // Create the Date object.
    var funcDateObject = new Date();
    // Returns the current date as a string.
    trace(funcDateObject);
    // Delete has no effect.
    delete funcDateObject;
    // Still returns the current date.
    trace(funcDateObject);
    // Set the object reference to null.
    funcDateObject = null;
    // Returns null.
    trace(funcDateObject);
}
// Call func() function.
func();
```

For more tips and techniques for creating content for mobile phones and devices, see www.adobe.com/go/learn_cs_mobilewiki_en.

Loading data for mobile devices in Flash Lite

When developing files for mobile devices, minimize the amount of data you attempt to load at one time. If you are loading external data into a Flash Lite file (for example, using `XML.load`), the device's operating system may generate a "memory failure" error if insufficient memory is allocated for the incoming data. This situation can occur even if the total amount of remaining memory is sufficient.

For example, suppose a file attempts to load an XML file that's 100 KB, but the device's operating system has allocated only 30 KB to handle that incoming data stream. In this case, Flash Lite displays an error message to the user, indicating that not enough memory is available.

To load large amounts of data, group the data in smaller pieces—for example, in several XML files—and make several data-loading calls for each piece. The size of each piece of data, and therefore the number of data-loading calls you need to make, varies by device and file. To determine an appropriate balance between the number of data requests and the likelihood of a memory failure, test files on a variety of target devices.

For optimum performance, avoid loading and parsing XML files if possible. Instead, store data in simple name/value pairs and load the data from a text file using `loadVars` or from precompiled SWF files.

For more tips and techniques for creating content for mobile phones and devices, see www.adobe.com/go/learn_cs_mobilewiki_en

Exclude classes from compilation for Flash Lite

To reduce the size of a SWF file, consider excluding classes from compilation but retaining the ability to access and use them for type checking. For example, try this if you are developing a file that uses multiple SWF files or shared libraries, especially those that access many of the same classes. Excluding classes helps avoid duplicating classes in those files.

- 1 Create a new XML file.
- 2 Name the XML file `FLA_filename_exclude.xml`, where `FLA_filename` is the name of the FLA file without the `.fla` extension. For example, if the FLA file is `sellStocks.fla`, the XML filename must be `sellStocks_exclude.xml`.
- 3 Save the file in the same directory as the FLA file.
- 4 Place the following tags in the XML file:


```
excludeAssets>
  <asset name="className1" />
  <asset name="className2" />
</excludeAssets>
```

The values specified for the name attributes in the `<asset>` tags are the names of classes that should be excluded from the SWF file. Add as many as required for the file. For example, the following XML file excludes the `mx.core.UIObject` and `mx.screens.Slide` classes from the SWF file:

```
<excludeAssets>
  <asset name="mx.core.UIObject" />
  <asset name="mx.screens.Slide" />
</excludeAssets>
```

For more tips and techniques for creating content for mobile phones and devices, see www.adobe.com/go/learn_cs_mobilewiki_en.

Index

Numerics

- 3D 139
 - Perspective Angle property 144
 - rotation 139, 142
 - translation 139, 140
- Vanishing Point property 146
- 3D Rotation tool 142
- 3D Translation tool 140

A

- absolute target path 180
- accessibility
 - Adobe Flash Accessibility web page 302
 - animation and 304
 - authoring for screens 323
 - automatic labeling 307
 - button and text field labels for 307
 - components 314
 - creating properties in ActionScript 312
 - creating reading and tab order using ActionScript 314
 - default reading and tab order 309
 - defining for entire Flash applications 308
 - descriptions for accessible objects 308
 - detecting screen reader with ActionScript 313
 - Flash Player and 303
 - hearing impaired users and 304
 - instance names and 306
 - keyboard navigation for 305
 - for movie clip children 308
 - naming buttons, text fields, and objects for 307
 - opaque windowless or transparent windowless modes and 303
 - options for Adobe Flash applications 309
 - screen readers 302
 - supported configurations 303
 - testing content 305
 - titles and descriptions for Flash applications 309
 - turning off automatic labeling 310
- Accessibility button, in Property inspector 308

- Accessibility panel
 - about 306
 - name vs. auto label 307
 - tab order 309
- actions
 - coding standards 420
 - deleting 338
 - keyboard shortcuts for 338
 - printing 333
 - reordering 338
 - selecting 338
- Actions panel 26
 - about 332
 - Actions toolbox 333
 - coding in 335
 - instance information in 156
 - Script navigator 333
 - Script pane 333
- Actions toolbox
 - adding an action with 338
 - resizing 333
 - viewing item descriptions in 338
 - yellow items in 341
- ActionScript
 - accessibility properties 312
 - class for screens 324
 - creating with Script Assist 337
 - detecting screen reader 313
 - editing preferences 336
 - formatting 339
 - in mobile devices 448
 - publish settings 371
 - screens and 327, 328
 - tab and reading order for screen readers 314
- ActionScript editing
 - check syntax 344
 - code hints 349
 - escape shortcut keys 342
 - find tool 343
 - importing and exporting scripts 344
 - line numbers 341
 - pin scripts 345
 - showing hidden characters 343
 - syntax highlighting 341
- activation of software 1

- ActiveX controls 380
- Adaptive color palette 394
- Add Shape Hint command 219
- adjust color filter, applying 241
- Adobe After Effects
 - integration with Flash 294
- Adobe AIR 398
 - application descriptor file 404
 - application files 399
 - installer files 399
 - signing applications 405
- Adobe Authorware, playing a Flash SWF file in 380
- Adobe ConnectNow 17
- Adobe Director, playing a Flash SWF file in 380
- Adobe Dreamweaver
 - integration with Flash 48
 - updating SWF files for 412
- Adobe Drive
 - using with Version Cue 49
- Adobe Exchange 4
- Adobe Fireworks
 - editing imported bitmap images with 88
 - importing files from 67
 - PNG files, importing 67
- Adobe Flash
 - creating mobile content in 48
 - integration with Adobe Premiere Pro 294
 - integration with After Effects 294
 - integration with Dreamweaver 48
 - integration with Illustrator 70
 - optimizing Flash content for mobile devices 445
 - quitting 47
- Adobe Flash Player
 - accessibility and 303
 - changing or uninstalling 381
 - configuring web server for 382
 - customizing context menu 376
 - debugging version 350
 - default reading order for screen readers 309
 - download simulation 445
 - file format 379
 - importing files into 65

- levels 177
 - printing from context menu of 52
 - security 382
 - text encoding 262
 - Unicode support 262
 - Adobe Flex 58
 - Adobe FreeHand
 - exporting files 415
 - Import Settings dialog box 66
 - importing files 65
 - importing with clipboard 125
 - Adobe Illustrator
 - about importing 69
 - compatibility with Flash 70
 - copy and paste 78
 - correcting incompatible effects 76
 - exporting AI files 413
 - import preferences 72
 - importing AI files 74
 - importing files into Flash 71
 - integration with Flash 70
 - Layers panel 76
 - object import options 76
 - Adobe Illustrator files
 - import preferences 72
 - importing 74
 - importing into the Flash library 72
 - Adobe Photoshop
 - compatibility with Flash 80
 - exporting files 411
 - import preferences 84
 - importing files 79
 - Adobe Premiere Pro
 - integration with Flash 294
 - Adobe Product Improvement Program 1
 - Adobe Version Cue
 - about 49
 - ADPCM compression, for sounds 279
 - Advanced effect, for symbol instances 154
 - After Effects. *See* Adobe After Effects
 - AI files. *See* Adobe Illustrator files
 - AI importer, Layers panel 76
 - AIFF sounds, importing 273
 - align attribute 389
 - Align panel 121
 - aligning
 - objects 121
 - text blocks 255
 - text characters 255
 - alignment, HTML (publish setting) 386
 - Alpha effect
 - instance property 154
 - partial transparency 393
 - ancestor screen, about 317
 - anchor points
 - adding 106
 - adjusting 107
 - converting between corner and curve 107
 - deleting 106
 - dragging 107
 - moving 106
 - nudging 106
 - showing on shapes 111
 - types of 93
 - animated GIF files
 - exporting 414
 - importing 65
 - publishing 393
 - animation
 - See also* animations, animation frames
 - accessibility and 304
 - classic tweens versus motion tweens 189
 - converting to movie clip symbol 150
 - creating keyframes in 211
 - displaying frames as onion skin outlines 221
 - dragging a library item onto a keyframe 211
 - editing frames in the Timeline 211
 - editing multiple frames 221
 - frame rates 184
 - frame-by-frame 220
 - frames in Timeline 184
 - graphics compared to movie clips 155
 - IK pose layers 230
 - inserting frames 211
 - inverse kinematics 223, 229, 230
 - linking layers to a motion path 215
 - modifying or deleting frames in the Timeline 211
 - motion paths for 213
 - motion presets 189
 - onion skinning 221
 - Play Once option 155
 - reversing the sequence of 211
 - saving 189
 - Single Frame option 155
 - unlinking layers from a motion path 215
 - animation, tweening
 - groups 212
 - instances 212
 - type 212
 - animations
 - in mobile devices 446
 - Antialias command 437
 - Anti-Alias For Animation option 245
 - Anti-Alias For Readability option 245
 - anti-aliasing
 - bitmaps 437
 - custom 245
 - device fonts 245
 - exported GIF 393
 - exported PNG 395
 - objects on clipboard 125
 - shapes 437
 - text 437
 - application descriptor file 404
 - applications
 - signing 405
 - area fill 133
 - array access operators, checking for matching pairs 344
 - arrow keys, moving objects with 124
 - Arrow tool. *See* Selection tool
 - Asian character entry on Western keyboard 265
 - audience for this document 28
 - Authorware. *See* Adobe Authorware
 - AutoCAD DXF files, importing 66
 - AutoCAD DXF Image 414
 - autoKeyNav parameter for slide screen 325
 - automatic labeling 309
 - accessibility, about 307
 - turning off 310
 - automating tasks 32
 - AVI files, exporting 417
- B**
- background color 45
 - Bandwidth Profiler
 - about 443
 - settings 444
 - Timeline graph 444
 - base attribute 389
 - behaviors
 - about 338
 - Bring Forward 169
 - Bring To Front 169

- controlling instances with 169
- Duplicate Movieclip 169
- GotoAndPlay at frame or label 169
- GotoAndStop at frame or label 169
- Load External Movie Clip 169
- Load Graphic 169
- screen navigation and control 326
- screen transitions 327
- Send Backward 169
- Send To Back 169
- Start Dragging Movieclip 169
- Stop Dragging Movieclip 169
- video, adding and configuring 298
- video, controlling video playback 297
- best practices
 - organizing Timeline 418
 - using scenes 178, 419
- bevel filter, applying 239
- bgcolor attribute or parameter 389
- Bind tool 229
- Bit Rate option, for mp3 sound
 - compression 279
- bitmap fills
 - applying 133
 - transforming 134
- bitmap images
 - anti-aliasing 39, 87, 437
 - breaking apart 89
 - compared to vector graphics 91
 - compressing as JPEG or PNG files 87
 - compressing for mobile devices 447
 - converting to vector graphics 89
 - editing 88
 - finding and replacing 61
 - importing 86, 125
 - in mobile devices 446
 - modifying filled areas 89
 - preserving transparency when
 - importing 63
 - setting compression options for 87
 - setting properties for 87
- Bitmap Properties dialog box 87
- Bitmaps on clipboard preference 39
- Blank Keyframe command 172, 211
- blending mode settings 242
- blending modes
 - about 242
 - applying 243
 - examples 243
- blends, in imported FreeHand files 66
- Blur filter 238
- BMP files
 - exporting 414
 - importing 65
- Bone tool 225, 226
- Break Apart command
 - bitmaps and 89
 - groups and 122
 - instances and 122
 - symbol instances and 155
 - text and 122, 252
- breakpoints
 - about 356
 - setting and removing 356, 361
 - setting and removing, in Actions
 - panel 356
 - XML file 357
- brightness
 - adjusting in Illustrator 11
- Brightness effect 154
- Bring Forward
 - behavior 169
 - command 121
- Bring To Front
 - behavior 169
 - command 121
- Brush tool
 - Lock Fill modifier 136
 - painting modes 102
 - painting with 101
 - setting brush size and shape 102
 - Wacom pressure-sensitive tablet 102
- button symbols 148
- buttons
 - accessible descriptions and labels for 308
 - adding sounds to 276
 - creating 164
 - disabling and enabling 165
 - disjoint rollover 165
 - Down state for 164
 - editing and testing 165
 - enabling 165
 - frame states for 164
 - Hit state for 164
 - input text fields, accessible labels for 307
 - naming for accessibility 307
 - Over state for 164
 - selecting enabled 165
 - testing 165
 - turning off accessible labels for 310
 - Up state for 164
- C**
 - call stack 362
 - center point 114
 - character position 255
 - child objects, making accessible 306
 - child screens
 - about 317
 - moving on the Stage 324
 - viewing 320
 - class name, for screens 328
 - class templates 56
 - class, document 373
 - classes
 - excluding from compilation 451
 - classid attribute 387
 - classpath 373
 - classpaths 374
 - Clear command 125
 - Clear Keyframe command 173, 211
 - Click Accuracy preference 97
 - clipboard
 - importing with 125
 - preferences 39
 - closing projects 54
 - code
 - displaying line numbers 341
 - formatting 339, 340
 - selecting a line 356
 - stepping through lines 357
 - word wrapping 341
 - code hints 347
 - manually displaying 348
 - specifying settings for 347
 - triggering 348, 349, 350
 - using 347
 - code stepping 362
 - codebase attribute 387
 - color palette
 - saving current as default 128
 - web-safe 128
 - color palettes
 - Adaptive 394
 - default 128
 - importing and exporting 128
 - color picker, opening 131

- colors
 - background 45
 - changing with the Property inspector 131
 - choosing for text 254
 - copying with the Eyedropper tool 133
 - creating and editing solid 129
 - default palette 128
 - default stroke and fill color, selecting 131
 - deleting 133
 - document background 44
 - duplicating 133
 - editing and creating solid 129
 - finding and replacing 60
 - importing and exporting palettes 128
 - in Actions toolbox 341
 - optimizing 437
 - saving current palette as default 128
 - selecting solid 131
 - selecting with Property inspector 131
 - setting maximum 394
 - sorting in Swatches panel 128
 - tweening 153
 - web-safe palette 128
 - commands
 - downloading 33
 - running 32
 - Commands menu
 - creating and managing commands 32
 - Edit Command List option 32
 - Get More Commands option 33
 - reusing commands 32
 - Run Command option 32
 - running commands 32
 - unrepeatable steps 33
 - Common Libraries submenu 160
 - Community help 2
 - Community search engine 2
 - compiled clip, in Library panel 157
 - compiler warnings 363
 - compiling conditionally 376
 - compiling FLA files 57
 - components
 - accessibility and 314
 - in Library panel 157
 - MediaController 298
 - MediaDisplay 298
 - MediaPlayback 298
 - screens and 329
 - components, creating for Flex 364
 - compressing sounds 278
 - compression
 - for mobile devices 447
 - Compression menu, for sounds 278
 - conditional compilation 376
 - config constants 376
 - configuration files 372
 - configuring a server for Flash Player 382
 - Connect Lines preference 97
 - Contact Sensitive option 97, 120
 - context menus
 - about 22
 - customizing in Flash Player 376
 - for screens 319
 - context sensitive Help 334
 - Control menu, Test Scene and Test Movie 444
 - Convert Lines to Fills command 113
 - Convert Stereo to Mono 279
 - Convert to Symbol command 149
 - Copy Frames command 173, 211
 - copying
 - history steps 31
 - layer folder contents 176
 - layers 175
 - objects 125
 - screens 321
 - Create Copy button, in Transform panel 125
 - curly braces, checking for matching pairs 344
 - curves
 - adjusting points and tangent handles 107
 - adjusting segments 107
 - dragging tangent handles on 107
 - drawing, with Pen tool 104
 - optimizing 112
 - reshaping 92
 - straightening and smoothing 111
 - Custom color palette 394
 - Custom Ease in or Ease Out dialog boxes 216
 - Custom option, for sound 275
 - Customize Shortcuts dialog box 42
 - Customize Toolbar command 21
 - Cut command 125
 - cutting a screen 321
- D**
- Darken blending mode 242
 - Debug Player 350
 - Debugger
 - buttons in 358
 - enable remote debugging 352, 364
 - Properties tab 354
 - selecting from context menu 353, 364
 - setting breakpoints 356
 - variables 353
 - Watch list 354
 - debugger for ActionScript 360
 - debugging 360, 362
 - Debug Player 350
 - files, protecting with password 384
 - from a remote location 352, 363
 - listing objects 355
 - listing variables 355
 - mode 361
 - remote files 363
 - text field properties 360
 - using the Output panel 358
 - variables 362
 - with trace statement 359
 - default color palette 128
 - Default Encoding preference 345
 - Default Text Orientation option 250
 - default workspace
 - restoring 15
 - Delete Workspace command 16
 - deleting
 - actions 338
 - frames or keyframes 173, 211
 - items, and saving documents 30
 - layers and layer folders 176
 - lines 113
 - objects 125
 - scenes 179
 - screens 322
 - deploying Flash SWF files 379
 - Deselect All command 119
 - Device Font publish settings 385
 - device fonts 245, 257
 - devicefont parameter 387
 - Difference blending mode 242
 - dimensions
 - default for document 44
 - publishing Flash SWF file 385
 - setting for document 44
 - direction lines and points, about 93
 - Director. *See* Adobe Director
 - display, speeding up document 437
 - distorting objects 115, 116

- Distribute to Layers command 186
- distributing
 - Flash SWF files 379
 - objects to layers 186
 - objects to top, bottom, left, right, or center 121
- dithering colors, GIF files 394, 396
- document class 373
- Document command 44
- document optimizing
 - colors 437
 - elements and lines 436
 - for playback 436
 - text and fonts 436
- document, setting
 - background color 44
 - dimensions 44
 - frame rate 44
 - properties for 44
 - ruler units 45
 - Stage size 44
- document-editing mode 19, 149, 150
- documents
 - Antialias display 437
 - background color, setting 44
 - colors, optimizing 437
 - context menu, custom 376
 - creating from template 44
 - deleted lines and items, removing and saving 30
 - dimensions, setting 44
 - elements and lines, optimizing 436
 - Fast display 437
 - Flash Player, loading into 177
 - form application 316
 - frame rate, setting 44
 - Full display 437
 - levels 177
 - loading into Flash Player 177
 - modifying 44, 45
 - opening new window 44
 - Outlines display 437
 - properties for, setting 44
 - Property inspector, modifying in 45
 - quitting, saving when 47
 - ruler units, setting 45
 - saving as template 46
 - saving Flash 46
 - saving in Flash MX format 46
 - saving when quitting 47
 - screen hierarchy 317
 - size report, generating 445
 - slide presentation 316
 - speeding up display 437
 - Stage size, setting 44
 - template, creating from 44
 - template, saving as 46
- Don't Replace Existing Items option 162
- dot syntax, target paths 180
- Down state for buttons 164
- Download Settings command 444
- download speed, for testing 444
- downloads
 - updates, plug-ins, and tryouts 4
- dragging objects 124
- Draw Border and Background option, for dynamic text 249
- drawing
 - anchor points 102, 107, 111
 - brush strokes 101
 - click accuracy tolerance 97
 - combining objects 118
 - converting lines to fills 113
 - Crop command 118
 - curve points and corner points 106
 - curves, optimizing 112
 - curves, smoothing 97
 - erasing lines or shapes 113
 - expanding shapes 113
 - fill edges, softening 113
 - interactive introduction 91
 - Intersect command 118
 - line end points, snapping 97
 - line segments, adjusting 107
 - lines, converting to fills 113
 - lines, straight 98
 - objects, snapping 123
 - ovals and rectangles 98
 - paths, with Pen tool 104
 - Pen tool 102
 - Pencil tool 101
 - pixels, snapping to 123
 - polygons and stars 101
 - precise lines and curves 102
 - Punch command 118
 - reshaping lines and shapes 111
 - rounded rectangles 100
 - shapes, modifying 113
 - showing anchor points on shapes 111
 - snapping line end points 97
 - snapping to objects and pixels 122
 - softening fill edges 113
 - stars 101
 - straightening and smoothing lines 111
 - tolerances for redrawing and straightening 97
 - Union command 118
- Dreamweaver. *See* Adobe Dreamweaver
- Drop Shadow filter 237
- Duplicate Movieclip behavior 169
- Duplicate Symbol command 150
- duplicating
 - symbols 150
- DXF Sequence, AutoCAD DXF Image 414
- dynamic text
 - about 244
 - HTML option 249
 - setting options 249
- dynamic text fields
 - accessible descriptions for 308
- E**
- easing 207
- Easing option
 - custom ease curve 216
 - for motion tweening 213
- edit bar 20
- Edit Envelope
 - for sounds 276
 - units in 277
- Edit in New Window command 151
- Edit in Place command 151
- Edit in Soundbooth 277
- Edit Multiple Frames button 221
- Edit Selected command 122
- Edit Symbols command 151
- editing
 - imported bitmap images 88
 - layers and layer folders 175
 - reshaping lines and shapes 111
 - softening edges of an object 113
 - symbols 150
 - text 248
- editing ease curves 210
- editing property curves 205
- editing property keyframes 206
- Effects menu, in the Property inspector 275
- Embed metadata 365

- embedded fonts
 - selecting 259
 - XML table 259
- enable remote debugging 352, 364
- Enable Simple Buttons command 165
- encoding text 344
- Enhanced Metafile files (Windows)
 - exporting 414
 - importing 65
- Envelope modifier 116
- EPS files
 - exporting 415
 - in imported FreeHand files 66
 - importing 65
- Erase blending mode 242
- erasing entire Stage 113
- error navigation 363
- errors in code 363
- Escape shortcut keys 342
- Event option, for sound 275
- event sounds 273
- Exchange 4
- Expand Fill command 113
- export file formats 411
- Export for Runtime Sharing option 163
- exporting
 - color palettes 128
 - F4V 295
 - FLV 295
 - scripts and language encoding 345
 - transparency 395
 - Windows Metafile files 415
- Extensible Metadata Platform 45
- external image editor, and imported bitmaps 88
- external library path 373
- Extras 4
- Eyedropper tool 133

- F**
- F4V 283, 285
 - exporting 295
- Fade options, for sound 275
- fading in or out 212
- Fast command 437
- file formats
 - alternative formats 379
 - exporting 411
 - importing 65
- file size
 - reducing for mobile devices 451
- files
 - importing 64
- fills
 - applying transparent 131
 - bitmap 89
 - copying 133
 - default color, selecting 131
 - edges, softening 113
 - expanding 113
 - gradient 130
 - lines, creating from 113
 - locked gradient or bitmap, with 136, 137
 - Paint Bucket tool, applying with 133
 - softening edges 113
 - swapping color with stroke color 131
 - text, for 255
- filters 239
 - about 234
 - animating 234
 - applying 235, 238, 239, 240, 241
 - Flash Player performance 235
 - Pixel Bender 235
 - saving preset settings 237
- Find And Replace
 - bitmap images 61
 - colors 60
 - fonts 59
 - overview 58
 - sound 61
 - text 59
 - video 61
- Find and Replace
 - screens 323
- Fireworks. *See* Adobe Fireworks
- FLA files
 - printing 50
 - saving 46
- Flash applications
 - naming for accessibility 308
- Flash content, aligning and cropping 386
- Flash Form Application 317
- Flash MX format, saving as 46
- Flash Player. *See* Adobe Flash Player
- Flash Project panel 52
- Flash Slide Presentation 317
- Flash SWF files
 - distributing 379
- Flash. *See* Adobe Flash
- FlashType
 - Anti-Alias For Readability option 245
 - custom anti-aliasing 245
- Flex 58
- Flex components 364
- Flex metadata 365
- Flip Horizontal command 117
- Flip Vertical command 117
- flipping objects 117
- FLV
 - exporting 295
- FLV files
 - dynamically playing external 296
 - exporting and importing 415
- folders
 - in Library panel 158
 - renaming, in projects 55
- font installation 2
- Font Mapping Default preference 38
- font symbol
 - identifier string for 251
 - Linkage option for 251
- fonts
 - choosing 254
 - creating font symbols 251
 - device 257
 - embedded and device 245
 - embedding 257
 - finding and replacing 59
 - mapping 253
 - optimizing 436
 - properties 254
 - selecting 254
 - selecting range of embedded 259
 - setting text attributes 254
 - Unicode 258
- form application 316, 317
- form screens 317
 - about 317
 - ActionScript class for 324
 - default visibility 325
 - document structure and 317
 - form application in 316
 - visible parameter 325
- formatting code 339, 340
- Frame by Frame graph, in Bandwidth Profiler 445
- Frame command 172, 211

- frame rate
 - in animation 184
 - setting 45
- Frame Rate option 44
- Frame View menu 23
- frame-by-frame animation 220
- frames
 - animation, editing in 211
 - animation, in Timeline 184
 - changing the view 23
 - converting keyframes into 173, 211
 - copying 173, 211
 - displayed in Timeline 22, 172
 - displaying as onion skin outlines 221
 - displaying contents 18
 - dragging in Timeline 211
 - editing in an animation 211
 - editing multiple 221
 - images, registering in 221
 - in mobile devices 448
 - inserting 172, 211
 - keyframes, converting into 211
 - onion skinning 221
 - pasting 211
 - printing 52
 - registering images in 221
 - removing 173, 211
 - showing thumbnails 24
 - testing performance with Bandwidth Profiler 444
- Frames button, in Edit Envelope 277
- Free Transform tool 115
- FreeHand. *See* Adobe FreeHand
- Full command 437
- Full Screen command 381
- FutureSplash Player files, importing 65

- G**
- Gap Size modifier, Paint Bucket tool 133
- General preferences, Disable Auto or Add Delete 106
- Generate Size Report option 445
- Get More Commands option 33
- GIF files
 - exporting 414
 - GIF89a file format 393
 - importing 65
 - publishing 393
- Global Positioning System (GPS) data 45
- glow filter, applying 239
- Goto command 180
- GotoAndPlay at frame or label behavior 169
- GotoAndStop at frame or label behavior 169
- GPS information in files 45
- gradient bevel filter, applying 241
- gradient colors 130
- gradient fills
 - applying 133
 - creating or editing 130
 - in imported FreeHand files 66
 - importing and exporting 128
 - working with solid colors and 129
- gradient glow filter, applying 240
- gradient pointers 130
- Gradients on clipboard preference 39
- graphic object, converting to a symbol 158
- graphic symbols
 - about 148
 - controlling with behaviors 169
- graphics
 - creating symbol instances 152
 - load behavior 169
 - setting animation options 155
- grayscale images
 - in imported FreeHand files 66
- grids 20
- grids, about 20
- Group command 121
- groups
 - breaking apart 122
 - creating 121
 - editing 121
 - locking 119
 - selecting 119
- guide layers 177
- Guided option 215
- guides
 - using 19

- H**
- H.264 283, 285
- Hand tool 19
- hardware acceleration 383
- height attribute 385, 387
- Help
 - context sensitive 334
- Hide Edges command 120
- Hide Screen option for child screens 320
- hiding and showing
 - objects from screen readers 310
- hierarchy, parent-child movie clips 178
- Highlight Color preference 39
- History panel
 - overview 30
 - Replay button 31
 - Save As Command option 32
 - saving commands from 32
 - screens with 322
 - steps, copying and pasting 31
 - steps, repeating 31
- Hit state for buttons 164
- horizontal text flow 248
- HTML
 - option, for dynamic text fields 249
 - publish settings 384
 - publishing templates 406
 - tag reference 380
 - templates 407
- HTML Alignment publish setting 386

- I**
- icons
 - above Script pane 335
 - in Debugger 358
 - panels as 14
- identifiers, assigning to sounds 87
- Illustrator. *See* Adobe Illustrator
- images
 - importing 63, 64
- import
 - SWF 296
- Import command 64
- Import For Runtime Sharing option 163
- importing
 - bitmap images 86
 - bitmaps with transparency 63
 - color palettes 128
 - files into the current Flash document 64
 - FLV files 292, 415
 - FreeHand and Fireworks PNG files 63
 - QuickTime 4 supported formats 65
 - scripts, and language encoding 345
 - sequence of files 64
 - sounds 273
 - TGA files 65
 - TIFF files 65
 - vector or bitmap files 65
- #include action in multilanguage text 271
- indenting code 340
- indenting text 255

- Info panel
 - instance information in 156
 - moving objects using 124
- Ink Bottle tool 132
- input text fields
 - accessible descriptions for 308
 - naming for accessibility 307
 - turning off accessible labels for 310
- input text, about 244
- Insert Blank Keyframe command 172, 211
- Insert Keyframe command 172, 211
- Insert Layer command 174
- Insert Target Path button 182
- instance names
 - and accessible objects 306
 - for screens 328
- Instance Properties dialog box 153
- instances, symbol
 - about 147
 - behaviors 155, 169
 - breaking apart 122
 - color and transparency, changing 153
 - creating 152
 - Info panel 156
 - information, getting 156
 - naming 152
 - properties, changing 153
 - swapping 154
 - unlinking from symbol 155
- interlacing
 - GIF files 393
 - JPEG files 395
 - PNG files 395
- Internet Explorer 380
- inverse kinematics 223
 - ActionScript 3.0 232
 - animation 230
 - armatures 223
 - Bind tool 229
 - Bone tool 225, 226
 - bones 223
 - bones in shapes 226
 - bones in symbols 225
 - easing animation 232
 - edit bones 227
 - motion constraints 229
 - pose layers 230
- Invert blending mode 242
- IPTC 45
- J**
 - JavaScript, alert statement 359
 - JPEG deblocking 383
 - JPEG files
 - importing 65
 - publishing 394
- K**
 - Kerning 255
 - keyboard controls
 - and Test Movie 351
 - in accessible content 305
 - keyboard shortcuts
 - adding and removing 42
 - coding in ActionScript 311
 - creating 311
 - customizing 41
 - deleting 42
 - for actions 338
 - for dialog box controls 35
 - for library items 36
 - for panels 33, 34
 - for pinned scripts 346
 - for Property inspector 33, 34
 - for the Stage 35
 - for tree controls 36
 - Shortcut option, on Accessibility panel 306
 - Keyframe command 172, 211
 - keyframes
 - animation, frame-by-frame 220
 - converting into frames 173
 - creating 211
 - creating blank 172, 211
 - dragging in tweened frame sequences 211
 - duration, extending 211
 - frame-by-frame animation 220
 - frames, converting into 211
 - inserting 172, 211
 - motion tweening 213
 - removing 173, 211
 - roving 197
 - selecting between two 119
 - using 172
- L**
 - languages, using multiple in scripts 345
 - Lasso tool
 - Magic Wand modifier 89
 - Magic Wand Settings modifier 89
 - Polygon mode 119
 - selecting objects with 120
 - Layer command 174
 - layer folders
 - changing order of 175
 - copying contents of 176
 - creating 174
 - deleting 176
 - editing 175
 - locking 176
 - organizing 175
 - renaming 175
 - layers
 - about 173
 - Add Layer button 174
 - adding layer folder 174
 - changing height of 24
 - changing order of 175
 - changing outline color 177
 - copying 175
 - creating 174
 - deleting 176
 - editing 175
 - guide layers 177
 - guided 215
 - hiding and showing 176
 - locking 176
 - mask 221
 - masking additional layers 222
 - organizing 175
 - renaming 175
 - selecting 119, 175
 - unlinking masked layers 223
 - viewing 176
 - learning interactions
 - system requirements for 301
 - Left Channel option, for sound 275
 - levels
 - absolute path 180
 - in Flash Player 177
 - naming in target path 180
 - libraries
 - assets, resolving conflicts between 161
 - common 160
 - components in 157
 - creating permanent 160
 - included in Flash 160
 - opening from other Flash files 157
 - resolving conflicts between assets 161

- sounds in 274
 - using shared 162
- library items
 - keyboard shortcuts for 36
- Library panel
 - about 26
 - adding an item to a document 158
 - deleting items in 159
 - displaying 158
 - finding unused items in 159
 - folders, using in 158
 - graphic object, converting to a symbol 158
 - options menu 158
 - using 157
 - using an item in another document 158
 - using folders in 158
- library path 373
- Lighten blending mode 242
- line numbers in code 341
- line numbers in code, displaying 341
- line spacing 255
- Line Style dialog box 131
- Line tool 98
- Linear Gradient option 130
- lines
 - converting to fills 113
 - drawing, with Pen tool 104
 - Eraser tool, removing with 113
 - modifying with the Ink Bottle tool 132
 - removing with Eraser tool 113
 - reshaping 92
 - selecting connected 119
 - selecting style and weight 131
 - straightening 111
- Link option, for text 253
- Linkage option
 - for font symbol 251
 - for sounds 87
- linking text blocks 253
- List Objects command 355
- List Variables command 355
- Live Effects, enabling or disabling 236
- Load Default Colors option 128
- Load External Movie Clip behavior 169
- Load Graphic behavior 169
- Load Order option 383
- loaded documents, controlling 182
- loading data in mobile devices 451
- loadMovie action and levels 178
- Lock command 119
- Lock Fill modifier 136
- locking layers 176
- loop attribute or parameter 388
- Loop option
 - about 155
 - for sound 275
- LOOP parameter publish settings 385
- looping
 - in accessible content 304
 - of animation sequences 155
- lossless compression, for bitmaps 87
- M**
- MacPaint files, importing 65
- Magic Wand modifier, for Lasso tool 89
- magnification (zooming) 18
- main toolbar 20
- Make Child Objects Accessible option
 - about 306
 - movie clips and 308
- Make Movie Accessible option 309
- Make Object Accessible option 306, 308
- margins, text 255
- mask layers
 - about 221
 - creating 222
 - linking additional layers 222
- Match Contents option 44
- Match Printer option 44
- Max Colors option 394
- MediaController component, about 300
- MediaDisplay component, about 300
- MediaPlayback component 300
- memory
 - managing in mobile devices 450
- menu attribute or parameter 390
- MENU parameter publish settings 385
- metadata 45
- Metadata panel 45
- metadata, Flex 365
- MIME types
 - configuring for 382
 - Flash Player 382
- Mixer panel 129
- mobile devices
 - optimizing Flash content for 445
- Modify Onion Markers button 221
- Motion Editor 203
 - easing tweens 207
 - edit ease curves 210
 - edit property curves 205
 - editing property keyframes 206
 - property curves 203
- Motion Guide command 214
- motion path
 - creating and hiding 213
 - linking layers to 215
 - orienting or snapping tweened elements to 214
 - unlinking layers from 215
- motion paths 195
 - paste stroke as 197
 - transforming 196
- motion presets 189
 - creating previews 192
 - exporting 191
 - importing 191
- motion tweening
 - about 212
 - along a path 213
 - linking and unlinking layers to a motion path 215
 - Motion Tweening option 212
- motion tweens
 - copy and paste properties 201
 - copy as ActionScript 3.0 202
 - creating 192, 193
 - easing 207
 - editing ease curves 210
 - editing property keyframes 206
 - motion paths 195
- movie clips
 - accessibility for children 308
 - accessible descriptions for 308
 - changing properties in Debugger 354
 - child, about 178
 - controlling with behaviors 169
 - controlling with target path 182
 - creating symbol instances 152
 - listing objects 355
 - listing variables 355
 - nesting 178
 - parent, about 178
 - parent-child relationship 178
 - symbol instances, creating 152
 - symbols 148
 - Timelines in 178

- Movie Explorer
 - about 27
 - context menu 28
 - displaying symbol definition 156
 - filtering displayed items in 27
 - Find text box 27
 - instance information 156
 - instances in 156
 - options menu 28
 - for screens 320
 - selecting items in 28
- movie parameter 387
- mp3, sounds, importing 273
- MSAA (Microsoft Active Accessibility) 303
- multilanguage text
 - about 258, 266, 267
 - creating with Strings panel 263
 - encoding language 261
 - #include action and 271
 - non-Unicode external files and 261
 - overview 261
 - system.useCodepage property 261
 - text variables 272
 - XMLConnector component 272
- Multiline option, for dynamic text 249
- multiple languages, using in scripts 345
- Multiply blending mode 242

- N**
- Named Anchor on Scenes preference 39
- names, choosing for accessibility 307
- namespaces, config 376
- naming conventions, variables 349
- Netscape Navigator 380
- New command 44
- New Font option, in Library panel 251
- New From Template command 44
- New Symbol command 149

- O**
- objects
 - accessibility options, defining 308
 - aligning 121
 - bringing forward (to front) 121
 - coding standards 420
 - copying 125
 - copying when transforming 125
 - cutting (deleting from file) 125
 - distorting 115, 116
 - dragging 124
 - drawing order 121
 - erasing 113
 - flipping 117
 - grouping 121
 - hiding from screen readers 310
 - making accessible 306
 - matching size 121
 - modifying with Envelope modifier 116
 - pasting 125
 - restoring transformed 117
 - rotating 116
 - scaling 116
 - selecting 118, 119
 - selection highlighting 118
 - sending backward 121
 - sending to back 121
 - size, matching 121
 - skewing 117
 - snapping 123
 - stacking 121
 - transformed, copying 125
 - transformed, restoring 117
 - transforming freely 115
- On Launch preferences 39
- onion skin markers
 - changing display of 221
 - moving 221
- Onion Skin Outlines button 221
- onion skinning 221
- online resources 4
- opaque windowless mode, and accessibility 303
- Open As Library command 157
- Open command 44
- Optimize option 112
- optimizing
 - curves 112
 - documents 436
 - GIF colors 393
 - PNG colors 395
- organizing scripts
 - actions 420
 - attaching to objects 420
- Orient to Path option, for motion tweening 213
- outlines
 - changing color on layers 177
 - viewing layer contents as 176
- Outlines command 437
- Output panel 358
 - and trace statement 359
 - copying contents 359
 - displaying 359
 - List Objects command 355
 - List Variables command 355
 - options 359
- Oval tool 98
- Over state for buttons 164
- overlayChildren parameter 325
- Override Sound Settings option 383

- P**
- Page Setup command (Windows only) 50
- Paint Bucket tool
 - applying fills with 133
 - Gap Size modifier 133
- painting
 - closing gaps with the Paint Bucket tool 133
- palettes
 - See also* panels
- panel menu, displaying 11
- panels
 - See also* palettes
 - about 10
 - Accessibility. *See* Accessibility panel
 - Actions 26, 156
 - Align 121
 - collapsing to icons 14
 - docking 12
 - Flash Project 52
 - grouping 14
 - History. *See* History panel
 - Info. *See* Info panel
 - keyboard shortcuts 33
 - Library 26
 - Mixer 129
 - moving 13
 - Scene 179
 - showing and hiding 11
 - stacking 14
 - Strings. *See* Strings panel
 - Transform. *See* Transform panel
- parameters
 - entering in Actions panel 338
 - for screens 325
- _parent alias 180
- parent screen, about 317
- parent-child relationships 178

- parentheses, checking for matching pairs 344
- passwords
 - and remote debugging 352, 363
 - for debugging files 384
- Paste command 125
- Paste Frames command 173, 211
- Paste In Place command 125
- pasteboard, showing and hiding 18
- pasting
 - history steps 31
 - objects 125
 - screens 321
- paths
 - adjusting anchor points in 107
 - direction lines and points on 93
 - drawing, with Pen tool 104
 - reshaping 92
 - tweening along 213
- pausing (stepping through) code 357
- Pen tool
 - adjusting anchor points with 107
 - corner points 106
 - curve points 106
 - drawing curves 104
 - overriding switching 106
 - pointer 108
 - preferences 107
 - using 102
- Pencil tool
 - drawing with 101
 - smoothing curves 97
 - straightening lines 97
- Perspective Angle property 144
- Photoshop. *See* Adobe Photoshop
- PICT files
 - exporting 413
 - importing 65
- PICT Settings for clipboard preference 40
- Pin Script option in the Actions panel 346
- pinning scripts, about 346
- Pixel Bender 235
- pixel snapping 123
- play attribute or parameter 388
- play modes, graphic instances 155
- Play Once option 155
- PLAY parameter publish settings 385
- playhead, moving 24
- playHidden parameter 325
- playing Flash content 381
- plug-ins
 - in Adobe Store 4
- pluginspage attribute 387
- PNG files
 - exporting 413
 - importing 65, 67
 - PNG filter options 396
 - publishing 395
- PNG Import Settings dialog box 69
- point size, choosing 254
- Pointer tool, setting options for 97, 120
- Polygon mode, for Lasso tool 119
- PolyStar tool 101
- preferences
 - Bitmaps on clipboard 39
 - clipboard 39
 - Drawing Settings options 96
 - editing 40
 - Font Mapping Default 38
 - FreeHand Text on clipboard 40
 - general 39
 - Gradients on clipboard 39
 - Highlight Color 39
 - Named Anchor on Scenes 39
 - On Launch options 39
 - Pen tool 107
 - PICT settings for clipboard 40
 - Printing Options (Windows only) 39
 - PSD files 84
 - setting 37
 - Shift Select 39
 - Show Pen Preview option 108
 - Show Precise Cursors option 108
 - Show Solid Points option 108
 - Show Tooltips 39
 - Timeline options 39
 - Undo Levels 39
 - vertical text 250
 - warning 38
- Preferences command 38
- Premiere Pro. *See* Adobe Premiere Pro
- previewing with Publish Preview command 397
- Print Margins command (Macintosh only) 50
- printing
 - actions 333
 - from authoring environment 50
 - FLA files 50
 - Flash Player context menu 52
- Printing Options preference (Windows only) 39
- PrintJob class
 - addPage method 368
 - properties 368
 - send method 370
 - start method 367
 - using 367
- projectors
 - playing with stand-alone player 381
 - stand-alone movie 380
- projects 52
 - closing 54
 - creating 53
 - Flash Project panel 52
 - opening 54
 - opening files 55
 - Project panel 52
 - renaming 55
- properties
 - sound 275
 - symbol instance 153
- Properties tab, Debugger 354
- Property inspector
 - about 25
 - changing units in 124
 - font properties 255
 - instances, for 156
 - keyboard shortcuts 33
 - modifying document properties 45
 - moving objects 124
 - screens, for 323
 - sound properties 275
 - Stroke and Fill Color controls in 131
 - video, changing properties 292
- Protect from Import option 383
- PSD files
 - importing 79
 - Publish options 84
- Publish options, PSD files 84
- Publish Preview command 397
- publish settings
 - ActionScript 371
 - generating HTML 384
 - modifying 371
- Publishing
 - Adobe AIR 398
- publishing
 - about 45
- punctuation balance, checking for 344

Q

quality attribute or parameter 388
 Quality option, for mp3 sound compression 279
 QUALITY parameter publish settings 385
 QuickTime
 files, exporting 416
 images, importing 65
 movie 380
 movies, importing sound only 274
 QuickTime video
 exporting 295
 Quit command 47

R

Radial Gradient option 130
 Raw compression, for sound 279
 read me file 2
 reading order
 in Accessibility panel 309
 in ActionScript 314
 default in Flash Player 309
 Recognize Lines preference 97
 Recognize Shapes preference 97
 Rectangle tool
 about 98
 Round Rectangle modifier 100
 Redo command 29
 redoing steps with the History panel 30
 registering images from frame to frame 221
 registration of software 1
 registration point
 changing 151
 displaying coordinates 156
 relative target path 180
 remote debugging 352, 363
 Remove Frame command 173, 211
 Remove Gradients option 393, 396
 removing
 screens 322
 renaming
 layers and layer folders 175
 projects or project folders 55
 reordering actions 338
 Repeat command 29
 repeating
 commands 32
 steps 29

replacing

 bitmap, sound, or video files 61
 colors 60
 fonts 59
 text 59
 reshaping lines and shapes 111
 resizing objects 115, 116
 Resolve Library Items dialog box 161
 restoring transformed objects 117
 Reverse command, for animation 211
 Revert command 46
 RGB colors, importing and exporting 128
 rich media templates 62
 Right Channel option, for sound 275
 Rotate and Skew command 117
 Rotate option, for motion tweening 213
 rotating
 clockwise or counterclockwise 117
 by dragging 117
 by 90° 117
 objects 116
 rotation, 3D 139
 round-trip editing
 sound 277
 roving keyframes 197
 Ruler Units menu 45
 rulers
 about 19
 running commands with Run Command 32

S

 align parameter 389
 SALIGN parameter publish settings 386
 Sample Rate
 for ADPCM sound compression 279
 for raw sound compression 279
 Save and Compact command 30
 Save As command 46
 Save As Template command 46
 Save command 46
 Save Current command 16
 Save Workspace command 16
 saving
 documents 46
 documents as templates 46
 removing deleted items and 30
 scale attribute or parameter 389
 Scale option, for motion tweening 213
 SCALE parameter publish settings 386

scaling

 by dragging 116
 objects 116
 Scene panel 179
 scenes
 best practices 178, 419
 changing order of 179
 pasting into 125
 selecting everything on every layer of 119
 testing download performance 444
 viewing 180
 working with 178, 179
 Screen Outline pane
 about 319
 expanding and collapsing 319
 selecting screens in 320
 showing and hiding 319
 screen readers
 about 302
 creating reading and tab order for 309
 default reading and tab order 309
 detecting with ActionScript 313
 hiding objects from 310
 Screenblending mode 242
 screens
 ActionScript and 327, 328
 ActionScript class, changing 324
 adding at the same level 318
 adding new screens 318
 ancestor screen, editing 321
 authoring accessibility and 323
 authoring environment 316
 Auto Snap option, for registration point grid 325
 behaviors for navigation and control 326
 child screens 317, 320, 324
 choosing type 318
 class name 328
 Class Name, in the Property inspector 325
 components and 329
 contents, editing 321
 context menu 319
 copying or cutting 321
 default screen and instance names 322
 deleting 322
 document structure and hierarchy 317, 319
 document types 316
 Document window, viewing in 320
 dragging and dropping 322

- editing contents 321
- Find and Replace 323
- first, adding 318
- Flash Form Application 317
- Flash Slide Presentation 317
- form screens 317
- fully rendering content 321
- instance name 323, 328
- Movie Explorer and 320
- moving 322
- moving a child screen on the Stage 324
- multiple, selecting 321
- naming 322
- navigation and control, behaviors for 326
- nested 319
- new, adding 318
- parameters 325
- parent screens 317
- pasting 321
- Property inspector, using with 323
- redoing and undoing steps with 322
- registration point 324
- Screen Outline pane 319, 320
- sibling screen 318
- slide screens 317
- top-level screen 317
- transition behaviors 327
- tree view 319
- type, choosing 318
- undoing and redoing steps with 322
- width and height, viewing 323
- workflow 316
- x* and *y* coordinates 324
- Script Assist, Actions toolbox 337
- Script navigator 332
- Script pane
 - adding actions in 338
 - buttons above 335
 - moving statements in 338
 - resizing 333
- Script window
 - about 332, 334
 - about breakpoints XML file 357
 - coding in 335
- scripts
 - correcting text display problems 344
 - importing and exporting 345
 - keyboard shortcuts for pinned scripts 346
 - pinning in place 345
 - searching 338
- scrolling text 252
- Seconds button, in Edit Envelope 277
- security
 - Flash Player 382
- Select Screen dialog box 326
- Selectable option
 - for dynamic text 249
 - for text 257
- selecting
 - adding to a selection 119
 - connected lines 119
 - deselecting 119
 - freehand selection area, with 119
 - hiding selection edges 120
 - keyframes, everything between two 119
 - Lasso tool, with 120
 - layers 175
 - locking groups or symbols 119
 - objects 118
 - objects, Contact Sensitive option 97, 120
 - scene, everything in 119
 - selection marquee, with 119
 - straight-edged selection area, with 119
 - text and text blocks 248
- selection highlighting, for objects 118
- Selection tool
 - reshaping with 111
 - selecting objects with 119
 - Smooth modifier 112
 - Straighten modifier 112
- Send Backward
 - behavior 169
 - command 121
- Send To Back
 - behavior 169
 - command 121
- shaders 235
- shape hints, for shape tweening 219
- shapes
 - copying 125
 - erasing 113
 - expanding 113
 - flipping 117
 - grouping 121
 - modifying 113
 - pasting 125
 - recognizing and redrawing 97
 - reshaping with the Selection tool 111
 - rotating 116
 - scaling 116
- selecting 118
- showing anchor points on 111
- skewing 117
- snapping 123
 - tolerance for redrawing geometric 97
- shared libraries
 - adding sounds to 87
 - font symbols 251
 - using assets 162
- shared library assets
 - during authoring, about 162
 - during run time 162
 - updating or replacing during authoring 164
- Show All command 18
- Show Frame command 18
- Show Grid command 20
- Show Pen Preview preference 108
- Show Precise Cursors preference 108
- Show Shape Hints command 219
- Show Solid Points preference 108
- Show Tooltips preference 39
- Show Warning Messages option 387
- sibling screen 318
- Silicon Graphics files, importing 65
- Simulate Download command 444
- Single Frame option 155
- Single Line option, for dynamic text 249
- size report 445
- skewing
 - objects 117
 - with Transform panel 117
- slide presentations 316, 317
- slide screens
 - about 317
 - ActionScript class for 324
 - default behavior when hidden 325
 - default navigation 325
 - document structure and 317
 - parameters 325
- Smooth Curves preference 97
- Smooth modifier, for Selection tool 112
- smoothing curves and lines 111
- Snap option, for motion tweening 213
- snapping
 - to objects and pixels 122
 - to pixels 123
 - to grid 20
 - tolerance, setting for objects 97
- Soften Fill Edges command 113

- software
 - activation 1
 - downloads 4
 - registration 1
- sound
 - round-trip editing 277
- Sound Designer II files, importing 274
- Sound object, using a sound with 87
- Sound Properties dialog box 278
- sounds
 - in accessible applications 302
 - ADPCM compression 279
 - buttons, adding to 276
 - compressing 278
 - efficiently using 280
 - envelope lines 277
 - envelopes, editing 276
 - event and stream 273
 - Event synchronization option 275
 - file size, tips for reducing 280
 - finding and replacing 61
 - importing 273
 - in library 274
 - looping 275
 - looping to reduce file size 280
 - options menu 275
 - properties 275
 - raw compression 279
 - reusing to reduce file size 280
 - shared libraries, adding to 87
 - Sound Properties dialog box 278
 - start point, setting 276
 - Start synchronization option 275
 - stop point, setting 276
 - Stop synchronization option 275
 - stream and event 273
 - stream synchronization 275
 - synchronizing 275
 - testing 279
 - Time In control 276
 - Time Out control 276
 - tips for reducing file size 280
 - versions, creating separate 383
- source path 373
- Special instance color property 154
- spell checking 250
- src attribute 387
- stacking objects 121
- Stage
 - displaying entire 18
 - erasing 113
 - keyboard shortcuts for selecting 35
 - size of 44
 - zooming 18
- stand-alone Flash Player 381
- Start Dragging Movieclip behavior 169
- Start option, for sound 275
- statements
 - reordering 338
 - trace 359
- static text
 - about 244
 - changing to dynamic text for accessibility 310
 - and screen reader reading order 314
- stepping code 362
- stepping through lines of code 357
- Stop Dragging Movieclip behavior 169
- Stop option, for sound 275
- Straighten modifier, for Selection tool 112
- straightening curves, lines 111
- Stream option, for sound 275
- streaming
 - sounds 273
 - testing performance 444
- Streaming Graph, in Bandwidth Profiler 445
- Strings panel
 - about 263
 - adding strings 264
 - automatic language detection 266
 - changing Stage language 265
 - default language 266
 - editing text 265
 - importing XML file 270
 - publishing 266
 - selecting languages 263
 - translating text 269
 - XML file format 267
- strokes
 - converting to fills 113
 - copying 133
 - default color, selecting 131
 - Ink Bottle tool, modifying with 132
 - line style, selecting 131
 - Selection tool, selecting with 119
 - swapping color with fill color 131
 - transparent, applying 131
- weight, selecting 131
- width of 131
- Subselection tool
 - adjusting line segments 107
 - showing anchor points 111
- substitute fonts
 - deleting 254
 - specifying 253
 - turning off alert 254
 - viewing 254
- Sun AU files, importing 274
- Swap Symbol dialog box 154
- Swatches panel
 - Add Colors option 128
 - loading default 128
 - Replace Colors option 128
 - Save Colors option 129
 - sorting 128
 - Web 216 option 128
- SWC files 364
- SWF
 - import 296
- SWF files
 - configuring for server MIME type 382
 - editing in Flash from Dreamweaver 48
 - exporting from Illustrator 71
 - frame load order 383
 - importing 65
 - JPEG compression 383
 - looping 385
 - playing 385
 - preventing import 383
 - printing frames 52
 - shortcut menu 385
 - substituting system fonts 385
- swliveconnect attribute 388
- symbol workflow 71
- symbol-editing mode 19, 148, 149, 151
- symbols
 - about 147
 - button 148
 - buttons, creating 164
 - converting a graphic object 158
 - creating 148
 - duplicating 150
 - editing 150
 - font 251
 - graphic 148
 - instance properties 153
 - instances, creating 152

- instances, unlinking from 155
 - movie clip 148
 - swapping 154
 - symbol-editing mode 151
 - tweening colors 212
 - types 148
 - unlinking from instance 155
 - viewing definition 156
 - Sync option, for sound 275
 - synchronizing
 - sounds 275
 - syntax
 - checking 344
 - color options, setting in the Actions panel 341
 - System 7 sounds, importing 274
 - system.useCodepage property 261
- T**
- tab order
 - about 309
 - in Accessibility panel 309
 - in ActionScript 314
 - default for accessibility 309
 - viewing 310
 - tangent handles, adjusting 107
 - target paths
 - about 181
 - absolute 180
 - expression 182
 - inserting 346
 - level names 180
 - relative 180
 - specifying 182
 - targetPath function 182
 - templates
 - creating 407
 - creating document from 44
 - publishing 406
 - rich media 62
 - sample 409
 - shorthand variables 409
 - variables 407
 - Test button, in Sound Properties dialog box 279
 - Test Movie command 165, 351, 444
 - Test Scene command 165, 444
 - testing
 - accessible content 305
 - Generate Size Report option 445
 - movies 351
 - sounds 279
 - text
 - aliasing 244
 - alignment 255
 - anti-aliasing 437
 - breaking apart 122, 252
 - character options 255
 - clipboard, importing with 125
 - color, choosing 254
 - device fonts 245, 257
 - dynamic text options 249
 - editing 248
 - embedded fonts 245
 - encoding 345
 - fields 244
 - fill color 255
 - finding and replacing 59
 - fixed width or height 248
 - flow, horizontal or vertical 248
 - font and paragraph attributes 254
 - font substitution 253
 - font symbols, creating 251
 - font, selecting 254
 - horizontal or vertical flow 248
 - importing from Illustrator into Flash 72
 - importing with clipboard 125
 - in mobile devices 446
 - linking to a URL 253
 - making selectable by users 257
 - margins 255
 - masking 246
 - multilanguage 258, 259, 261, 266, 267
 - optimizing 436
 - point size, choosing 254
 - properties, choosing 254
 - resizing a text block 248
 - scrolling 252
 - searching for in scripts 338
 - selecting 248
 - selecting a font 254
 - selecting device fonts 257
 - spell checking 250
 - style, choosing 254
 - text fields 244
 - transforming 252
 - translating in Strings panel 269
 - Unicode in Flash Player 262
 - URL, linking to 253
 - widening text block 248
 - text blocks
 - appearance 247
 - resizing 248
 - selecting 119, 249
 - widening 248
 - text fields
 - displaying properties for debugging 360
 - naming for accessibility 308
 - text fonts
 - choosing 254
 - device 245
 - embedded 245
 - outlines 245
 - properties 254, 255
 - selecting 254
 - selecting device 257
 - substituting missing 253
 - symbols, creating 251
 - text report, in HTML file 410
 - text variables, using in multilanguage text 272
 - TGA files, importing 65
 - TIFF files, importing 65
 - Time In control, for sounds 276
 - Time Out control, for sounds 276
 - Timeline
 - absolute target path 180
 - alias, parent 180
 - animation frames in 184
 - appearance of, changing 23
 - best practices 418
 - converting keyframes into frames 173
 - copying and pasting frames 173, 211
 - deleting frames or keyframes 173
 - docking to the application window 23
 - dragging 23, 211
 - editing 211, 221
 - frame display, changing 23
 - frames, deleting 211
 - frames, inserting 172
 - frames, onion skinning 221
 - frames, using 172
 - hiding layers in 176
 - keyframes 172
 - keyframes, converting into frames 211
 - keyframes, creating in 211
 - keyframes, deleting 211
 - layer folder order, changing 175
 - layer height, changing 24
 - layer name fields in 23

- locking layer folders in 176
- locking layers in 176
- in movie clips 178
- multiple timelines 177
- organizing 418
- parent alias 180
- playhead 24
- Preview option 24
- relative target path 180
- resizing 24
- showing frame thumbnails in 24
- target paths 181
- using 22, 172
- viewing layers as outlines 176
- working with frames 22, 172
- Timeline frames, about 211
- Tint effect 154
- Tint instance property 154
- tolerance, for snapping to objects 97
- tool tips. *See* code hints
- toolbar, customizing 21
- tools
 - Brush 101
 - customizing the toolbar 21
 - Eraser 113
 - Eyedropper 133
 - Free Transform 115
 - Hand 19
 - Ink Bottle 132
 - Lasso 120
 - Line 98
 - Oval 98
 - Paint Bucket 133
 - Pen 102
 - Pencil 101
 - Pointer 97, 120
 - PolyStar 101
 - Rectangle 98
 - selecting 21
 - Selection 119
 - Subselection 106
 - viewing 20
 - Zoom 18
- Tools panel, configuring 11
- Tools panel, using 20
- Trace Bitmap command 90
- Transform panel
 - copying objects with 125
 - skewing objects with 117
 - undoing transformations with 118
- transformation point 114
- transformations
 - combining 115
 - pointers 115
- transforming
 - objects 125
 - text 252
- transitions
 - motion tweening 212
 - for screens 327
- translation, 3D 139
- transparency
 - adjusting separate color values 154
 - alpha 154
 - exporting 395
 - partial 393
 - preserving in imported bitmap images 63
 - tweening 153
- transparent windowless mode, and
 - accessibility 303
- tree controls, keyboard shortcuts for 36
- troubleshooting. *See* debugging
- tryouts 4
- tween spans
 - copy and paste 201
 - editing 198
- tweened animation 187
- tweened frames, dragging keyframes in 211
- tweening
 - along a path 213
 - motion 212
 - motion paths for 214
 - shape 183, 218
 - symbol colors 212
- text
 - See also* type
- U**
- Undo
 - button, in Transform panel 118
- Undo command 29
- Undo Levels preference 39
- undoing steps
 - with the History panel 30
 - and redoing 29
 - and redoing, with screens 322
- undoing transformations 117
- Ungroup command 121
- Unicode
 - about 261
 - Flash Player support 262
 - font selection 258
 - support 345
- unpinning scripts in the Actions panel 346
- Up state for buttons 164
- Update button, in Sound Properties dialog
 - box 278
- updates 4
- updating Flash SWF files for
 - Dreamweaver 412
- updating sounds 278
- URLs, listing in HTML file 410
- UTF-16 BE and UTF-16 LE 262
- UTF-8 262
- UTF-8 (Unicode) 345
- V**
- Vanishing Point property 146
- Variable option for dynamic text 249
- variables 362
 - and Debugger Variables tab 353
 - and Debugger Watch list 354
 - HTML template 407
 - modifying in Debugger 353
 - naming 349
- Variables tab, Debugger 353
- vector graphics
 - compared to bitmaps 91
 - creating from imported bitmap images 89
 - importing with clipboard 125
 - in mobile devices 446
- version control
 - best practices 419
- Version Cue. *See* Adobe Version Cue
- vertical text
 - flow 248
 - preferences 250
- video
 - behaviors, adding and controlling 297, 298
 - components 298
 - exporting QuickTime 295
 - F4V 283, 285
 - finding and replacing 61
 - FLV files, importing 292
 - FLV files, playing external 296
 - H.264 283, 285
 - playback, controlling 292
 - properties of, changing 292

- tips for creating 286
- updating embedded video 292
- View Esc Shortcut Keys command 338
- View menu, changing document display with 437
- View Options pop-up menu 341
- visible parameter, for form screen 325

W

- Wacom pressure-sensitive tablet 102
- warning preferences 38
- warping objects 116
- Watch tab, Debugger 354
- WAV sounds
 - exporting 417
 - importing 273
- Web 216 color palette 394
- web servers, configuring for Flash Player 382
- Web Snap Adaptive color palette 394
- web-safe color palette 128
- weight, for lines 131
- Welcome screen 17
- width attribute 385, 387
- window, opening new 44
- windows
 - moving 12
- Windows Metafile files
 - exporting 415
 - importing 65
- wmode attribute or parameter 390
- word wrapping in code, enabling 341
- work area
 - about 10
- workspaces
 - customizing 15

X

- XFL 294, 296
- XFL files 47
- XLIFF 267
- XML files 267, 270
- XMLConnector component
 - multilanguage text 272
- XMP metadata 45

Z

- z axis 139
- Zoom tool 18
- zooming 18