

Adobe **Flash** CS5 Intermediate

Instructor: Tom Corcoran

Course **Outline**

Day One

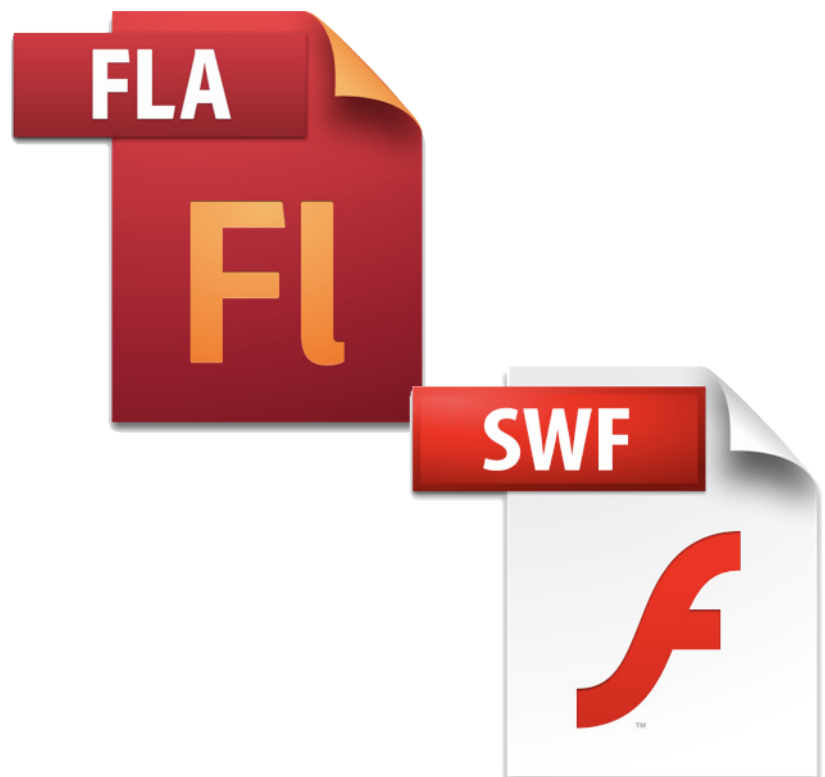
- i. Adding Armature (chapter five)
 - a. using the bone tool with symbols
 - b. constraining movement
 - c. animating armatures using symbols
 - d. using the spring property
 - e. animating armatures using shapes
 - f. runtime vs. authoring armatures
- ii. Break
- iii. Adding Text (chapter seven)
 - a. adding static text to the stage
 - b. formatting text
 - c. Dynamic and Input text
 - d. Text Layout Framework (TLF)
 - e. embedding Text in Flash

Day Two

- i. Interactivity (chapter six)
 - a. creating buttons
 - b. sound effects on buttons
 - c. duplicate and swap symbols
- ii. Break
- iii. ActionScript Basics
 - a. what is ActionScript?
 - b. writing ActionScript
 - c. labeling keyframes
 - d. using Code Snippets

Day Three

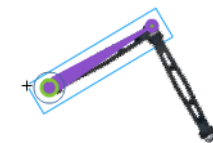
- i. Audio and Video (chapter eight)
 - a. import and edit sound files
 - b. Adobe Media Encoder
 - c. play external video
 - d. video playback components
 - e. cue points
 - f. alpha levels in video
 - g. embedding video in Flash
- ii. Break
- iii. Publishing Flash (chapter ten)
 - a. Publish settings
 - b. HTML and SWF
 - c. Flash Player Versions
 - d. Self-contained Projector files



Adding Armature

Bone Tool

This tool allows you to “link” symbols together to animate the symbols along joints (similar to the way a human elbow functions). Place your symbols on the same layer on the same keyframe. Then using the Bone Tool click on a starting point of your symbol, drag the mouse to the location you want the symbols to link and release the mouse button. These linked symbols become what is called an Armature. This is just like a skeletal structure of the human body moving like elbows, knees or other joints and bones. You can repeat this process for as many joints and symbols as you have.



Constraining Joint Movement

These joints in the Armature rotate completely 360° and don't move vertically or horizontally by default. You can change this default by using the Properties Panel to constrain the Joint Rotation (think about how any human joint normally functions... joints are confined by a certain range of rotation rather than rotating all the way around) and/or you can allow a symbol/joint to move along a range of motion (X & Y).



Animating Armatures using Symbols

The new linked Armature structure will be placed on a new Armature layer. This layer of the timeline can be animated just like a Motion Tween layer. Move the playhead to any location along the Armature layer timeline and click and drag a part of the symbol to begin to control your Armature structure into a new pose. You can clear a pose by moving the playhead over that pose and right click the mouse and choose Clear Pose. This tool and armature animation makes animating a skeletal like structure much easier than motion tweening.

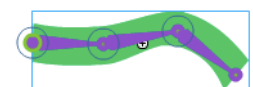
Using Spring Property of Armatures

Armatures can have a Speed, Spring and Dampening properties added to each or all of the bone structure. The lower the Speed value the more “weight” the bone has. Spring Strength has to do with the rigidity of the joint. A value of 100 is the least flexible (most rigid structure) whereas a value of 1 is most flexible (more like a wet piece of spaghetti). Dampening is the rate of decay of the Spring. Higher values of Dampening cause the Spring effect to decay quicker.



Animating Armatures using Shapes

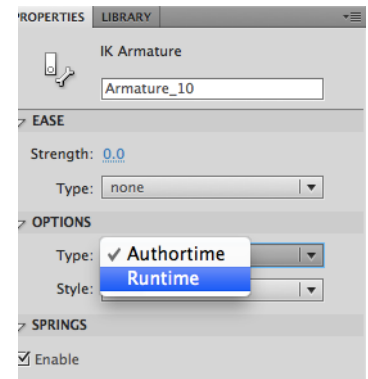
Armatures can also be placed on any shape you draw on the Stage using Flash's shape drawing tools. Shapes need to be in merge mode. Then the shape is animated on it's own Armature layer the same way symbols Armatures are animated.



Adding Armatures

Runtime vs. Authortime Armatures

Armatures can be set as single poses, changed to Runtime Armatures in the Properties Panel and tested as a .SWF. This file allows you to grab ahold of the end of a shape armature, click and drag the mouse and interact with the armature as the end user. Authortime Armatures however can have many poses, create a tween or motion effect to the whole armature but don't allow for end user interaction. Armatures default as Authortime armatures allowing the Flash designer to set poses and create animated motion over the timeline.

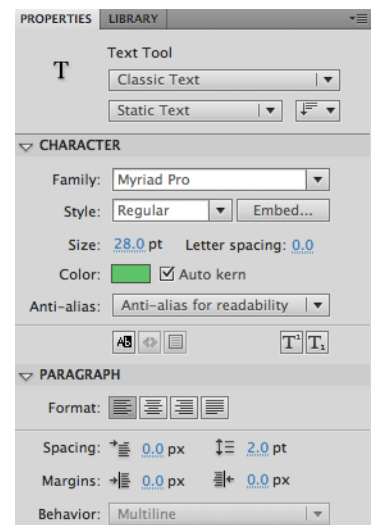


Adding Text

Adding Static Text to the Stage

Using the Text Tool you can click on the Stage and begin to type text or click and drag out the width size of the text box you'd like to begin entering text into and then begin to type. Classic Text can be added as Static Text which is defined in Flash as text you control by typing the characters straight into the text box. Static Text is opposed to Dynamic or Input Text which I discuss below.

T text tool

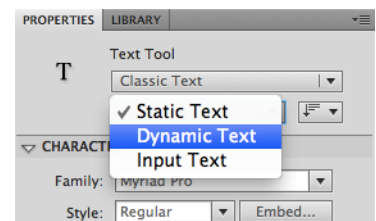


Formatting Text

Text formatting options include choosing a typeface, boldfacing, italicizing, the color of the text, paragraph formatting and indenting/spacing and more similar to any word processing program or even a page layout program like Adobe InDesign.

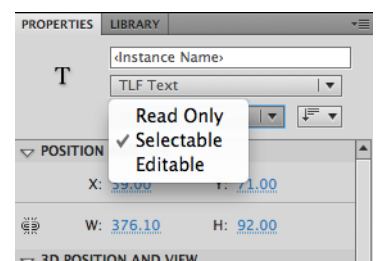
Dynamic and Input Text

Two other options for text boxes on your Stage are Dynamic Text or Input Text. Choosing Dynamic Text from the drop down menu which defaults to Static Text allows you to manipulate the text using ActionScript by giving the textbox an instance name to refer to. Input Text allows you to create a text box that your end users can fill in (ex. a contact form with areas for First Name, Last Name, Address, and so on). End users can enter their information into a Input Text Box and that information can be used by Flash or eventually passed to a database using a server side scripting language like JSP or PHP.



Text Layout Framework (TLF)

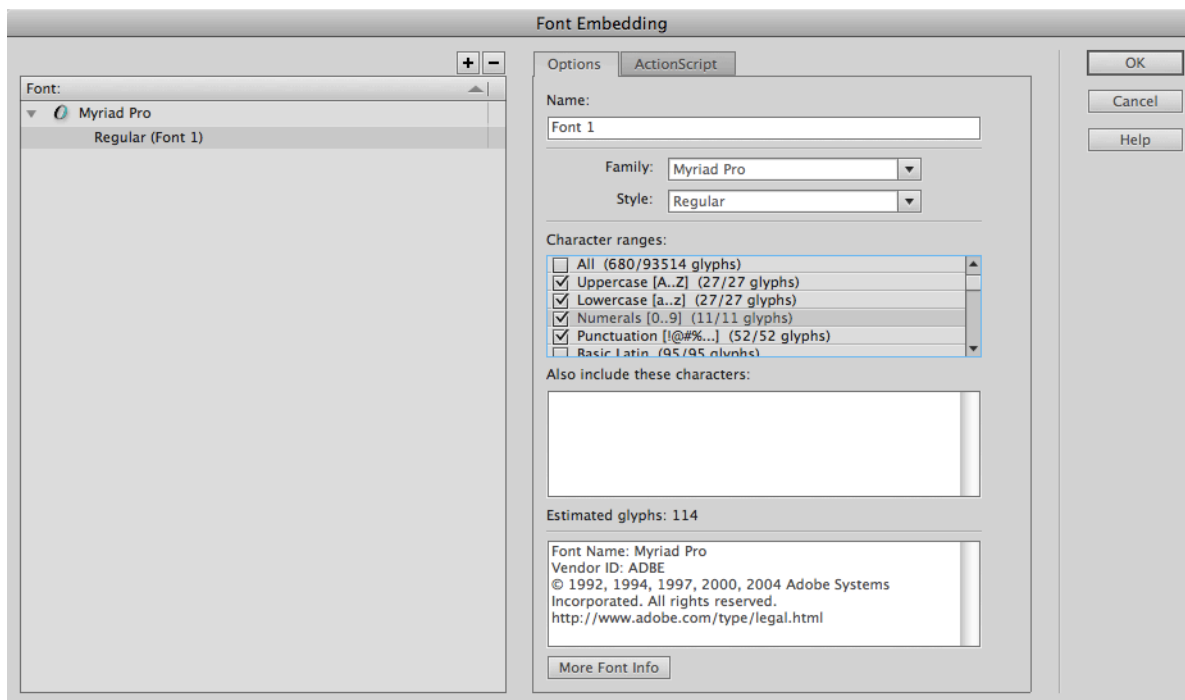
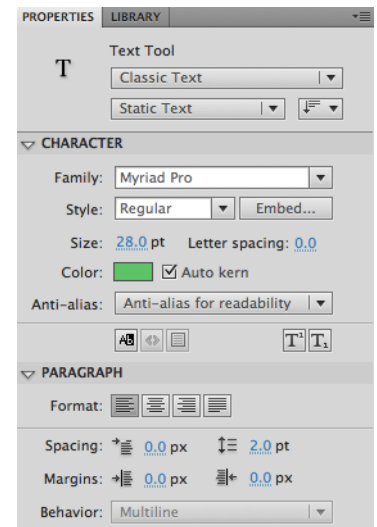
A new feature added to Flash CS5 is the Text Layout Framework (TLF). This is a new type of textbox has more features including layout of multiple textboxes on the screen and linking those boxes together similar to linking text boxes in a page layout program like Adobe InDesign. Linking textboxes so all overflow text flows from one textbox to another textbox gives you greater control over the flow of text and provides a nice feature to create more advanced text layouts.



Adding Text

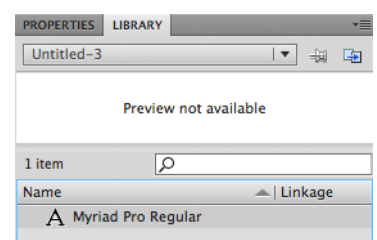
Embedding Text in Flash

In order for your final .SWF file to look consistent across different users computers and operating systems you may want to embed fonts to maintain the correct appearance. You don't need to embed fonts if you are using Static textboxes with the Anti-Aliasing property set to Use Device Fonts. Otherwise you will receive a message telling you that you will likely want to embed Fonts into your final .SWF file. You can embed fonts by doubleclicking into your textbox and highlighting some text. This will make sure that that typeface is chosen in the Font dropdown menu. Then click the embed button. This will open up the window for embedding fonts. You can type a name for this embedded Font, make sure the Font Family and Style match what you are trying to embed and then choose all of the Character Ranges to embed. Usually you'll want to embed the Uppercase, Lowercase, Numerals and Punctuation (unless you want All Character ranges including Latin, Arabic, Hebrew and more). When you are finished click OK.



Your embedded font(s) will be added to the Library of items to include in the final .SWF file and embedded so the text appearance remains consistent from user to user.

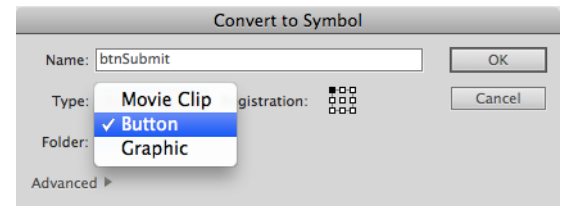
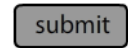
The more Fonts and/or Character ranges that you embed into a Flash project the larger your final .SWF file will be so keep file sizes in mind when embedding fonts.



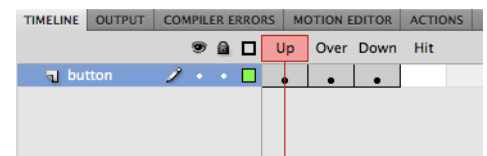
Interactivity

Creating Buttons

Buttons provide a way for your end users to interact with your Flash file. Buttons can be drawn in Flash or imported in as graphics from another program such as Adobe Illustrator or Photoshop. When you have the element on the screen you want to be a button select that shape/element on the screen and press F8 (or Modify->Convert to Symbol from the menu in Flash). Type in a symbol name and drop the menu down to choose Button as the symbol type. Then click OK.



Buttons drawn in Flash (or imported in as vector graphics from Illustrator) can be easily edited to include more “states” than just a simple button sitting on the screen if you choose. Double click on your button either in the Library or on the Stage to enter into the Button Editing Mode. Here you will see a 4 frame layer where you can create Up, Over, Down and Hit states for your button. Copy keyframes by choosing the next blank frame to the right of the Up state and press F6. Repeat this for Down and Hit states



Button States

Up: normal button appearance

Over: mouse over button appearance

Down: mouse clicked down button appearance

Hit: defines the clickable area of the button

Compatible Flash CS5 Audio File Types:

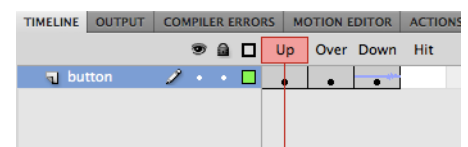
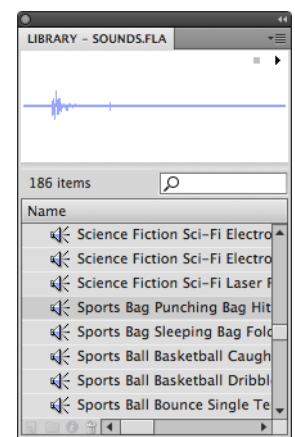
- .aac: advanced audio coding
- .aiff: audio interchange file (Macintosh only)
- .au: sun microsystems type
- .avi: audio video interleave
- .mp3: motion picture experts group level 3
- .wav: waveform audio

Interactivity

Sound Effects on Buttons

Buttons can have sound effects included on them so when the users rolls over or clicks down on the button they hear a sound effect play. A selection of sound effects is included with Flash CS5 under Windows->Common Libraries->Sounds or you can import your own audio sound into Flash from another source into the Library in your Flash file (see Compatible Flash Audio File Types). To Import Sound go to File->Import->Import to Library. My personal recommendation is to use .MP3.

Sound Effects are usually added to either the Over or Down button state. In the Button Edit Mode select the Over or Down state frame (which ever you want to add a sound effect to). Drag and drop the sound out of the Library (or Sounds Library) and onto the Stage. You will see a small blue “waveform” appear on that keyframe. That sound will play when you roll over or click down on that button after publishing or testing your .SWF file.

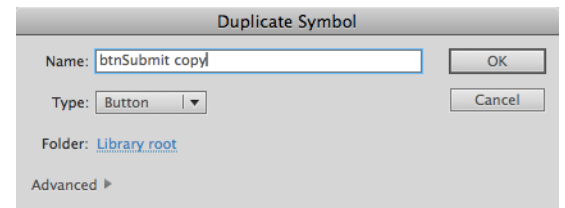
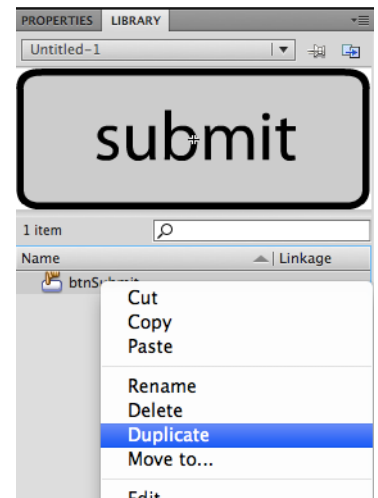


Interactivity

Duplicating and Swapping Symbols

In the Library in your Flash project you will end up with different types of bitmap images (JPG, PNG, GIF), symbols (movieclip, buttons, graphics), sound files, and video. Any time you want to create a clone or copy of any of those items you can do that by opening up the Library, right clicking on the item in the Library and choosing Duplicate. Flash will prompt you to give that item a new name. After naming it click OK and you will have an exact copy of that item in the library. If they item you duplicated was a symbol (movieclip, button, or graphic) you can double click on that new symbol and edit the size, shapes, colors or other properties without having to worry about editing the original symbol. This is highly useful for making multiple buttons that have the same shape, size and color but have different text labels on the button or movieclip symbols of a variety of sizes and colors but of the same basic shape.

Swapping symbols also can be useful. To swap out one symbol or bitmap image for another select the symbol/image on the Stage. In the Properties Panel click on the Swap button. This will allow you to pick a new symbol/image to swap out for the current symbol/image from your Library. Choose the new symbol/image and click OK and the new symbol/image will appear in the original symbol/image's place on the Stage replacing the original.



ActionScript Basics

What is ActionScript?

ActionScript is an Object Oriented Programming (OOP) language for developing more complex and/or interactive projects in Adobe Flash. ActionScript is currently in version 3.0 of this language. ActionScript was first introduced into Flash in Macromedia Flash 4 (released in 1999) and has developed into a more robust and complete scripting language. All Object Oriented Programming languages work in the same basic way (although they have different syntax to the code). Some common OOP languages include C#, C++ and Java.

OOP languages see the “world” very much the same way that human see and interact with the world. Everyday we interact with objects. Those objects do certain things (a chair allows you to sit in it) and have certain properties (the chair in my office is colored grey). OOP Languages and particularly ActionScript 3.0 also see and interact with objects which have methods and properties. You can have a red circle movieclip symbol on that Stage which represents the object. The red circle movieclip's color property is red. And as a movieclip this symbol has an independent timeline of it's own which can be played

In this handout ActionScript 3.0 code you type into the Actions Panel will look like this typeface:
`gotoAndPlay (2) ;`

To write a line of comments to yourself or another designer you type 2 forward slashes and type your comments:
`//comment goes here`

For a multiline comment type:
**`/* comments go here
second line here
*/`**

ActionScript Basics

What is ActionScript? (continued)





back (a method of movieclips is it's ability to be played back). So by this example we see that ActionScript can deal with objects which have methods (things they can do) and properties (features about that object like specific color or width and height values).

The following are the 8 basic parts of the ActionScript 3.0 language:

1. **Objects:** these are symbols or items you have on your Stage. If you have given that item an instance name you can refer to it by name using ActionScript. Movieclip symbols, Button symbols, and Video imported to the Stage or dragged in from the Library which can be given a name

2. **Methods:** these are actions that these objects can take or have applied to them. To stop the playhead from playing on the main timeline/Stage you invoke the `stop()` ; method. Another example is that Button symbols can “listen” for keyboard key presses or mouse clicks and other events by invoking the `addEventListener()` ; like this:

```
play_btn.addEventListener(MouseEvent.CLICK, playThis);
```

			
object: button instance	method invoked on button	specific event listener	call to a Function named playThis

3. **Properties:** aspects of objects such as size, shape, and position on the Stage. These values can be modified by first typing the object instance name, a period to access the property or method, and then the property to be modified. Then type equals and your new assigned value.

```
firstName_txt.text = "Tom";
```





		
dynamic text instance	text property	assigned value String "Tom"

```
minuteHand_mc.rotation = 45;
```

		
MovieClip Symbol instance	rotation property	assigned value Number 45

4. **Events:** these are items that Flash can “listen” for and then respond to once Flash detects that the event in question has occurred. A few examples of of Events that take place in Flash are mouse clicks on buttons, keyboard presses by the user, or even the detection of everytime the playhead enters a frame on the Stage.

```
play_btn.addEventListener(MouseEvent.CLICK, playThis);
```

			
object: button instance	method invoked on button	specific event listener	call to a Function named playThis

ActionScript Basics

What is ActionScript? (continued)

5. **Functions:** these are chunks of ActionScript code telling Flash something you want done as the result of an Event (something happening... like when the user clicks the play button you want the a movieclip playhead to start playing). The function name is your choosing.

function keyword	function name	function event name and type
<code>function</code>	<code>playThis</code>	<code>(myEvent:MouseEvent)</code>
		<code>:void{</code>
		<code>//this function causes the playhead to play</code>
		<code>play();</code>
		<code>}</code>

method invoked everytime this function runs

6. **Variables:** containers for values (ex. Text Strings, Numbers, or True and False). Variables are data typed meaning that you define what type of data that variable is going to contain. A few common data types are:

String: Text you want to store in ActionScript in Quotes (ex. "Heartland"),
Number: Decimal numbers (ex. 2.42),
int: Whole numbers (ex. 3630)

<code>var</code>	<code>currentDay</code>	<code>:String</code>	<code>=</code>	<code>"Monday"</code>	<code>;</code>
var keyword	variable name and Data Type	assigned String value of Monday			

7. **Arguments:** details passed to a method within the parenthesis that occur after you type the method name. For instance to have the playhead go to frame 24 and play type:

```
gotoAndPlay(24);
```

method invoked on button	argument applied to the method: frame 24
--------------------------	--

Mouse Events:

```
MouseEvent.CLICK  
MouseEvent.MOUSE_MOVE  
MouseEvent.MOUSE_DOWN  
MouseEvent.MOUSE_UP  
MouseEvent.MOUSE_OVER  
MouseEvent.MOUSE_OUT
```

Keyboard Events:

```
KeyboardEvent.KEY_UP  
KeyboardEvent.KEY_DOWN
```

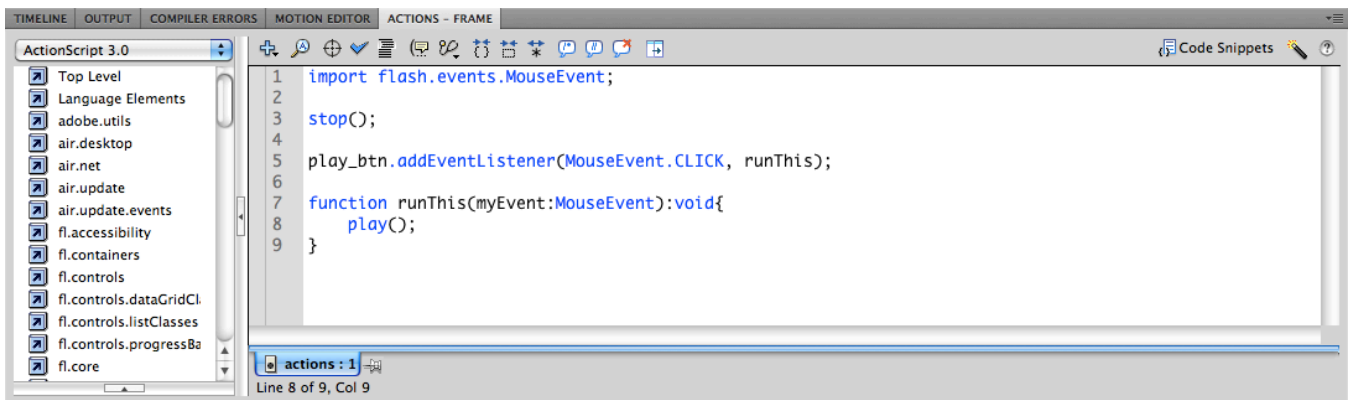
ActionScript Basics

What is ActionScript? (continued)

8. **Keywords:** words that ActionScript has assigned a certain definition. These words should be avoided when naming variables or functions. Keywords show up in blue in ActionScript. In the example above the word “var” is in blue because it is a keyword reserved in ActionScript for declaring a variable (so you wouldn’t want to call any of your symbol instance names “var” or name a variable the word “var”. You will learn what all of these keywords are as you use ActionScript. If a word appears in blue that you have assigned to a variable name or instance name it is recommended to change that name to something else so ActionScript isn’t confused.

Writing ActionScript

You will write all of your ActionScript 3.0 code onto a blank keyframe(s) on a separate layer (usually titled “actions”) in your Flash projects within a Panel called Actions (Window->Actions from the menu or the F9 key). So first thing you usually do when starting a new ActionScript 3.0 project is to create a new layer above Layer 1 and rename it **actions**. Then you have a blank keyframe ready to type your ActionScript code onto. The Actions Panel is composed of three main parts. First you will see Actions Toolbox to the left. This area contains all of the ActionScript divided out into categories depending on the usage of that piece of ActionScript code. Second the main section of the screen is a text entry area to type in your ActionScript code. This is the main area you will work in when adding ActionScript. Third there is a drop down menu in the upper right where you can set up this Actions Panel (set up Word Wrapping, Line Numbers, Print the code out, etc).



Labeling Keyframes

Frames can be labeled by using Blank Keyframes, selected that blank Keyframe and typing in a name in the Properties Panel. These labeled keyframes can then be referenced in ActionScript. So if you set a label on frame 10 of your timeline called “About” you can refer to that frame label name in ActionScript directing the playhead to that keyframe by name rather than by number.

ActionScript Basics

Labeling Keyframes (continued)

To refer to a labeled keyframe in ActionScript you would type something similar to this:

```
gotoAndStop ("About") ;
```

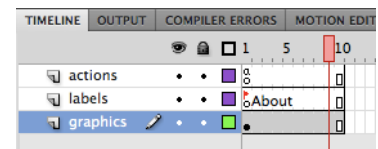
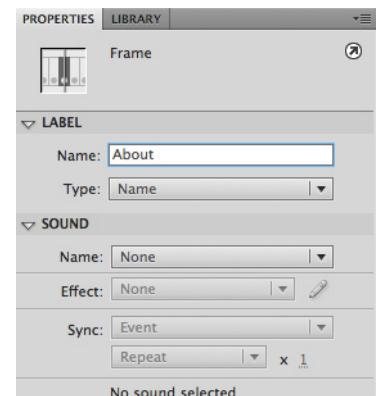
method
frame label location

This line of ActionScript would move the playhead directly to the keyframe labeled "About" and stop the playhead from playing forward.

To achieve this same idea without labeling the keyframe in this example you would type the following ActionScript using the keyframe number instead of a keyframe label:

```
gotoAndStop (10) ;
```

method
frame number location

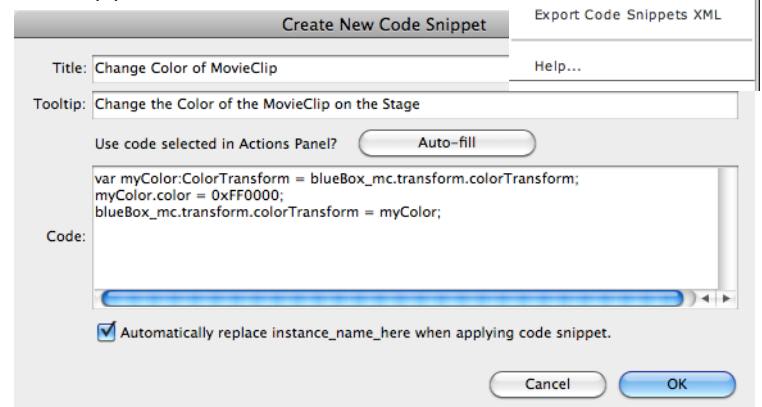
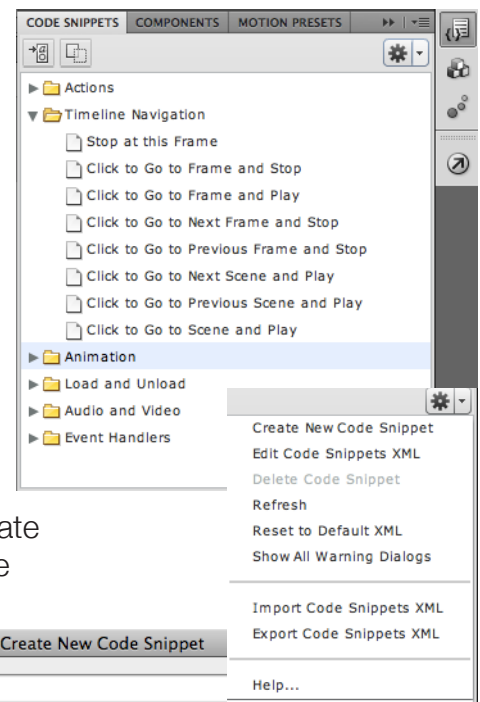


Using Code Snippets

A new feature of Adobe Flash CS5 is the Code Snippets Panel. This panel allows you to use preprogrammed pieces of ActionScript code to add to your own projects.

Some Code Snippets require to you select a movieclip symbol or button symbol on the stage to the snippet knows what you want the code applied to... but the ActionScript is still always written onto a Actions layer keyframe (not onto the symbol itself).

You can create a Code Snippet from your own ActionScript if you find that there is code you use often and want to save. To create your own Code Snippet highlight the code you want to copy in the Actions Panel, drop the menu down in the Code Snippets Panel shown at right. Choose Create New Code Snippet. Here you can type a Title, a Tool Tip (pop up description) and click Auto-Fill. The code you have highlighted in the Actions Panel will paste into the box. Click OK and your code will be added as a Code Snippet.



Audio and Video

Import and Edit Sound Files

Using File -> Import -> Import to Library you can bring files into your Flash project's Library Panel. Audio can be imported using this method. See page 5 of this handout for Audio File Types compatible with Flash. You can preview the audio sound file by clicking on the small play button in the upper right of the preview window in the Library Panel. Also preloaded sounds included with Flash can be found under:

Window -> Common Libraries -> Sounds.

Sounds can be placed in a Blank Keyframe on the timeline by creating a new Layer, and select the blank keyframe on that layer. Then go to your Library (or the Common Sounds Library) and drag the Sound file out of the library and onto the Stage. The only indication you get that a sound exists now on the Stage is up in the Timeline. A small blue waveform of that sound will show up in that keyframe. To hear that sound playback you will need to add frames to that layer (as many frames as the length of the sound file). Click the mouse down 20 or 30 frames from that keyframe and press F5 to add frames. You will see the blue waveform expand out to show the length of the sound file. If the sound doesn't playback entirely you'll need to add even more blank frames by going down further on that layer, selecting a frame and pressing F5 key again.

That sound file will then playback whenever the playhead reaches that keyframe with the sound. It will only playback when the playhead encounters the first frame of that sound (the original keyframe) and the playhead is moving.

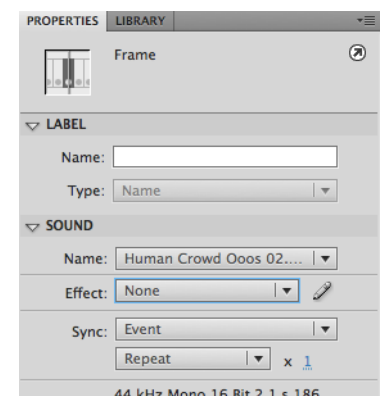
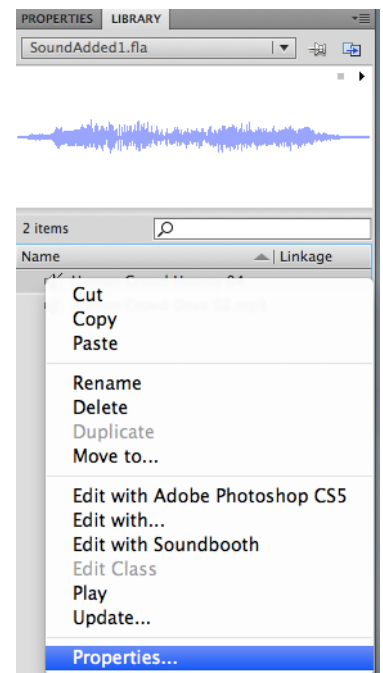
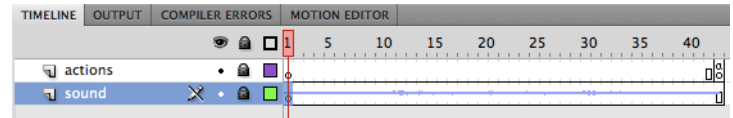
You can do some simple effects to the Sound once it is on a keyframe in Flash. Clicking on that keyframe that contains the sound will show properties of that sound in the Properties Panel. Here you will see the Name of the Sound (which can be swapped out for any other sound in your Library), Fade Effects and Sync Options.

Effects Options:

Left Channel and Right Channel: controls the speakers,
Fade to Right or to Left: Fades sound to one speaker,
Fade In and Fade Out: Fades sound in or out,
Custom: Allows greater control over volume, fades, and speakers.

Sync Options:

Event: triggers the sound to play,
Start: triggers the sound to play only if it isn't already playing,
Stop: stops the sound from playing,
Stream: sound syncs up with the timeline to match frames for animation.



Audio and Video

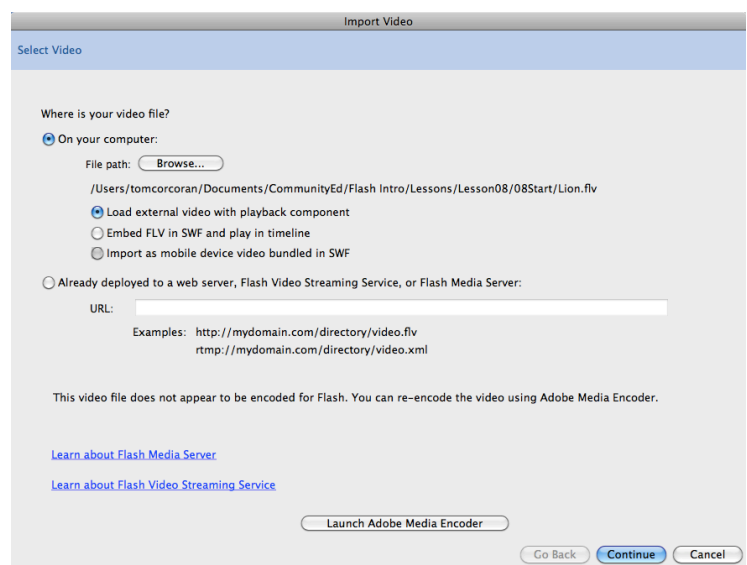
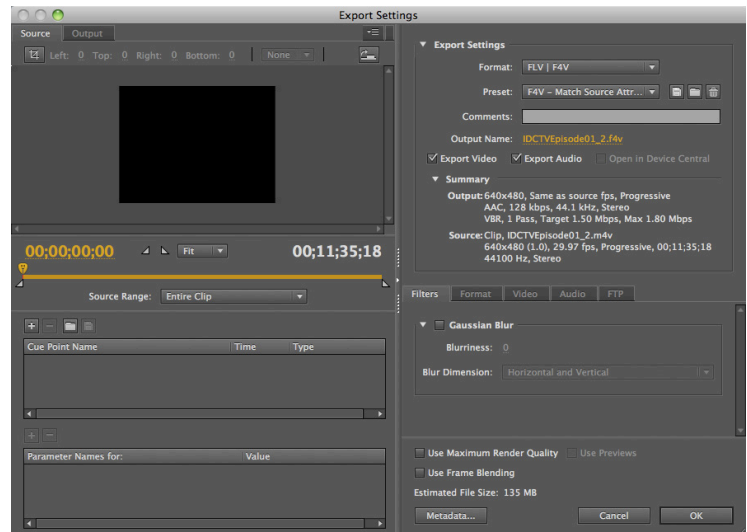
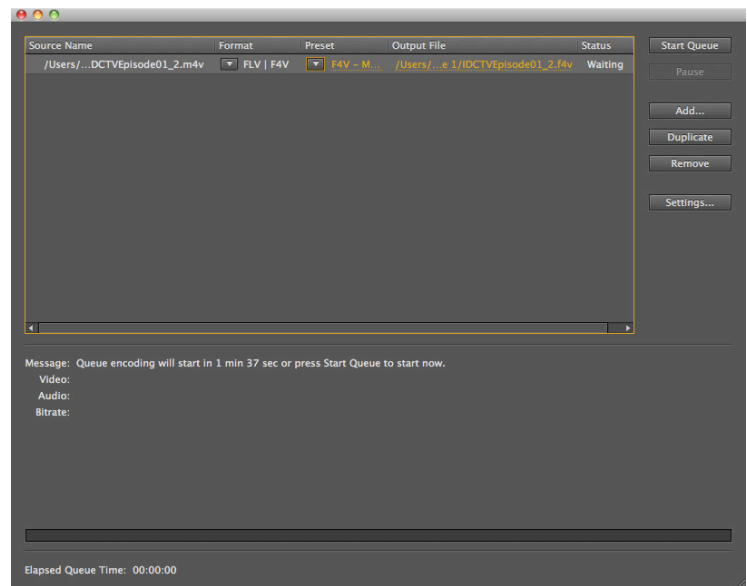
Adobe Media Encoder

Included with the installation of Adobe Flash CS5 is a separate program called Adobe Media Encoder. This program allows you to convert your videos to the appropriate format to include into your Flash project. Using the Adobe Media Encoder allows you to take a QuickTime file, or a Windows Media Video file and convert it into a Flash Video (.FLV) file type with precise settings for distribution over the web. You don't have to use the Adobe Media Encoder to import several different video file types but using it will help ensure the proper compression and sizing for your distribution method (web site, blog, CD-ROM, etc).

You start by launching the Adobe Media Encoder. From here you can drag and drop (or click add and browse to) the video(s) you want to encode. Clicking the Settings button will bring up the second screenshot shown at right allowing you to control virtually every aspect of your video including the format, presets for video size, audio quality and more. You can even trim start (in) points and ending (out) points of the video. Click OK when done with the Settings screen and click Start Queue to encode your video(s).

Play External Video

After creating a new Flash project you can import video. To do so you go to the File menu -> Import -> Import Video. This will launch the screen show at right. Here you can browse to the location of your video, choose to Load external video with playback components (recommended method) or Embed the video in the timeline (not recommended). Click Continue to advance to the next option screen.

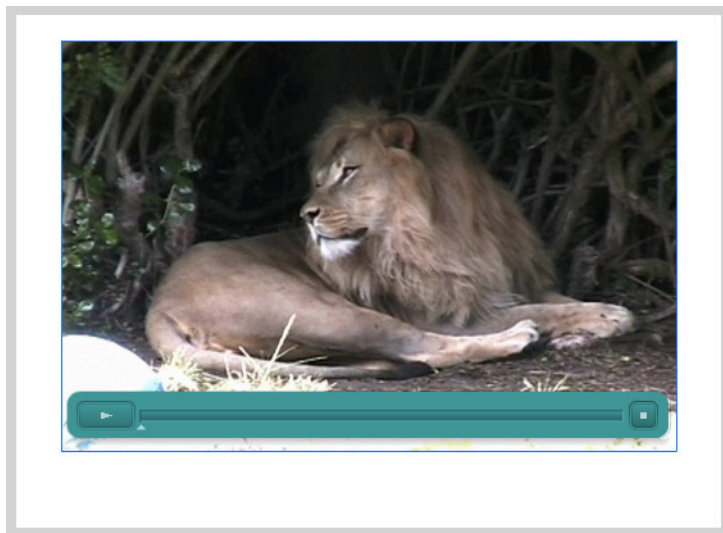


Audio and Video

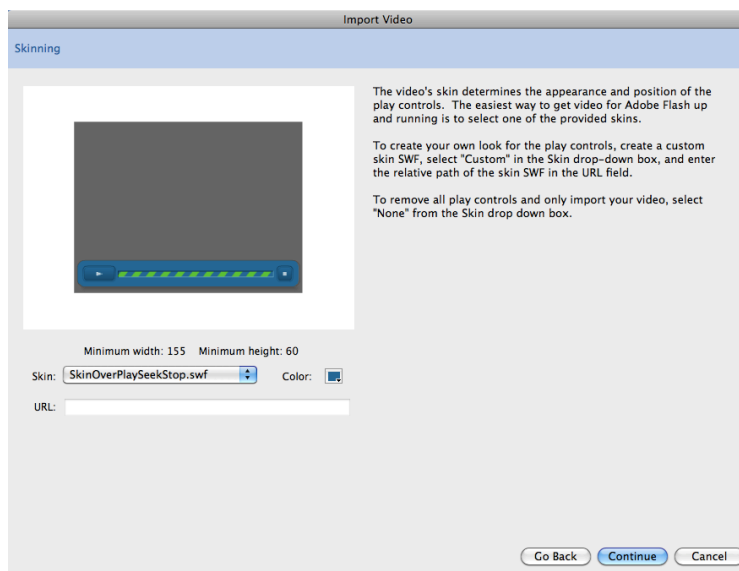
Video Playback Components

After choosing to Load external video with playback components, you are presented with a Skinning screen shown at right. This allows you to choose from many preset skins which allow you to add playback components (play button, stop button, pause button, mute button, volume controls, etc). This give you control over what playback components will appear either under or hover over the bottom of your video. You can also choose the color of the controls from this screen. Changing the Skin you choose can be done later on from the Properties Panel once the video in on the Stage in Flash.

Once you've chosen a color and appropriate skin for the video playback controls click Continue. Your video will appear as an item on the Stage in your Flash project file (as shown in the image below).

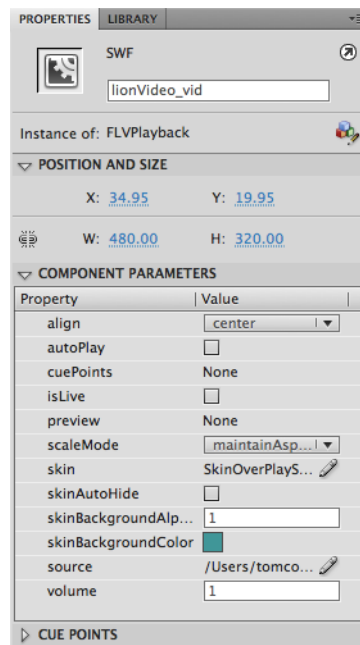


The Video Components properties of this Skin can be controlled or changed by selecting on the video on the Stage and looking at the Properties Panel (shown at right). Here you can change the skin, the skin background color, auto hide a skin so the mouse has to be rolled over the video to display the playback components and more. This video on the Stage acts just like any other item you add to a keyframe in the timeline. You can reposition it on the Stage, change the timeline location of the keyframe or even motion tween it.



Compatible Flash CS5 Video File Types:

- .avi: audio video interleave
- .flv: flash video
- .mov: quicktime movie
- .mp4: motion picture experts group layer 4 part 14
- .wmv: windows media video

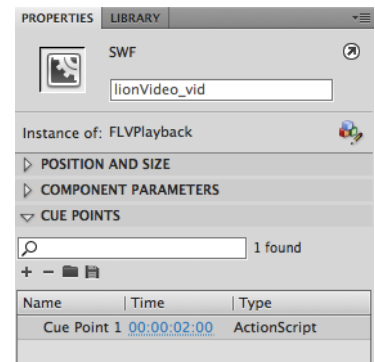


Audio and Video

Cue Points

Cue Points become “markers” you place into the video that you can use to have ActionScript control playback of the video. For instance if you have a 10 minute long video and 6 minutes into the video you have a break and start talking about a different topic, you could set a cue point at that 6 minute mark, then tell ActionScript that each time a certain button on the Stage is pressed to automatically jump the video playback up to the 6 minute mark and begin playing from there. This allows you to create scenes in the video (similar to a feature length film on DVD) or a table of contents of that video with text and buttons that will jump to certain cue points when clicked (like a Seek function on a DVD player).

Cue Points are added by selecting the video and opening the Properties Panel and open up the Cue Points option. This will allow you to click the plus (+) button to add a cue point. Give your Cue Point a name by double clicking the Cue Point 1 name and type a new Cue Point Name. ActionScript can then be written along with a button to jump to that Cue Point. There are Cue Point Code Snippets in that Panel for ActionScript.



Alpha Levels in Video

If your video was shot using a green screen and the file type your video is saved in allows for transparent background then Flash can utilize that transparency. The video you bring into Flash will have to be in the .FLV video format (Flash Video) and be encoded using the On2VP6 codec (the compression/decompression algorithm used to render/save the video). Flash will allow this video to have a transparent background which allows you to place the video over the top of a graphic or image background of your choice or simply have a white background behind the video.



Embedding Video in Flash

In addition to having Flash refer to a video saved elsewhere on your computer for playback, you can bring video into the timeline and playback just like you'd play a motion, classic, or shape tween. This is only recommended for very short video clips that have no audio. You will likely experience audio sync difficulty if you embed video into the timeline in Flash.

If you choose the recommended import option of Load external video with playback components (rather than Embed into the timeline) the video won't be embedded into the .FLA or .SWF Flash files. The .SWF is merely linked to the video file. This means that if you ever move your .FLA or .SWF file to a new location on your computer (or upload the .SWF to a web server) you will also have to move/copy the video to the new location as well or the video link will be broken.

Publishing Flash

Publish Settings

Under File->Publish Settings you can control what file types you publish out from your original Flash .FLA project file. Here you can set up options for the .SWF file you export, whether you want to also publish the file to HTML, create an animated .GIF or still .JPG or .PNG images, or create self-contained projector files for Windows PC or Macintosh platforms.

Checkmarking these boxes (other than Windows Projector or Macintosh Projector) will give you a options tab across the top of this Publish Settings screen. Clicking on these tabs will change out this screen and give you even more control over the creation of these types (shown below with the Flash tab chosen).

The Flash tab has many options for controlling all aspects of how your .SWF file will playback on the end-users computer.

Options include:

Player: the version of the Flash Player

Script: the version of ActionScript

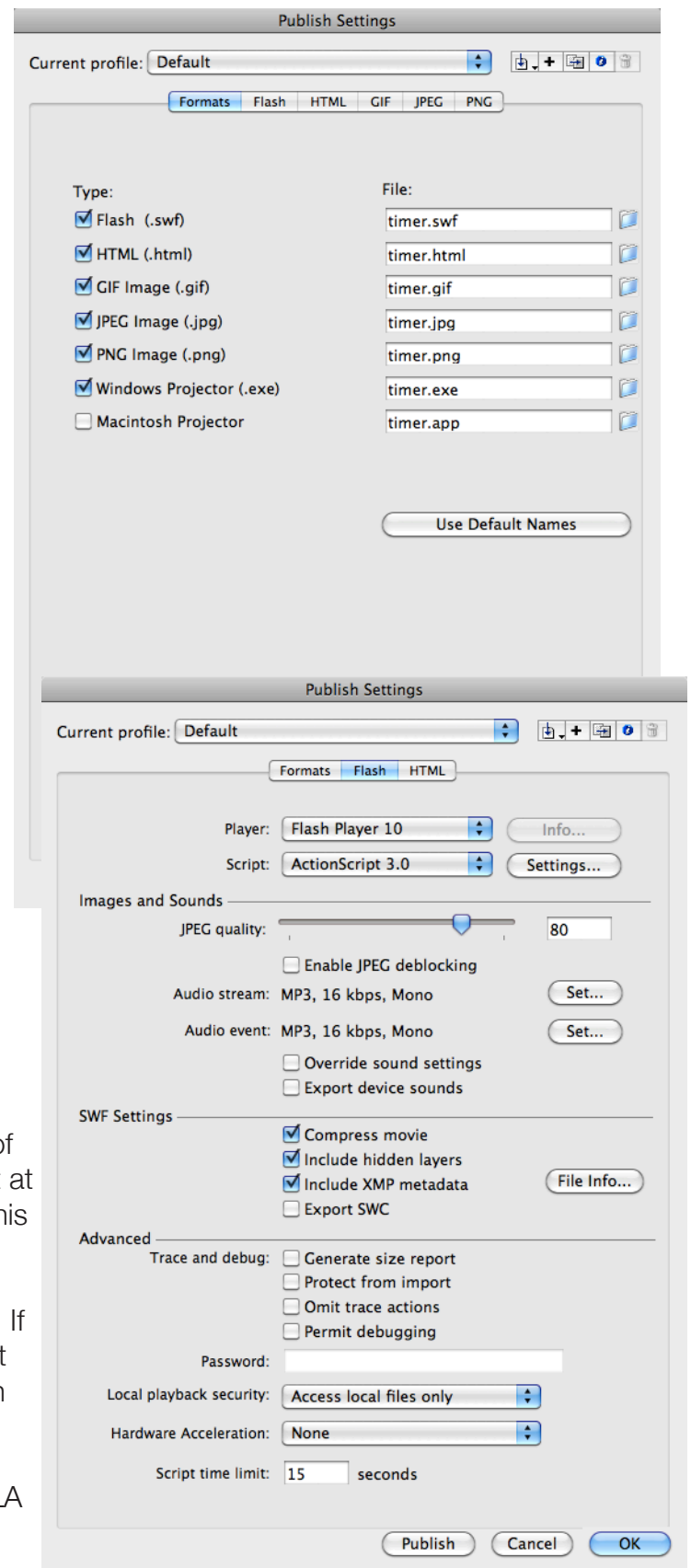
JPEG quality: image compression quality

Audio stream: audio options and quality

SWF Settings: Exporting options

Advanced Settings: debugging and other

Once you are finished setting up your publish settings simply click on Publish at the bottom of this screen. You can also Publish your files out at any time in Flash by pressing Shift and F12. This will publish all of the file types you checkmark and save them by default into the same folder your original Flash .FLA project file is stored in. If you want these files to be stored in an different location click on the Select Publish Destination folder (off to the right of the File name). This allows you to save any of these published files out to a different location than your original .FLA project file.



Publishing Flash

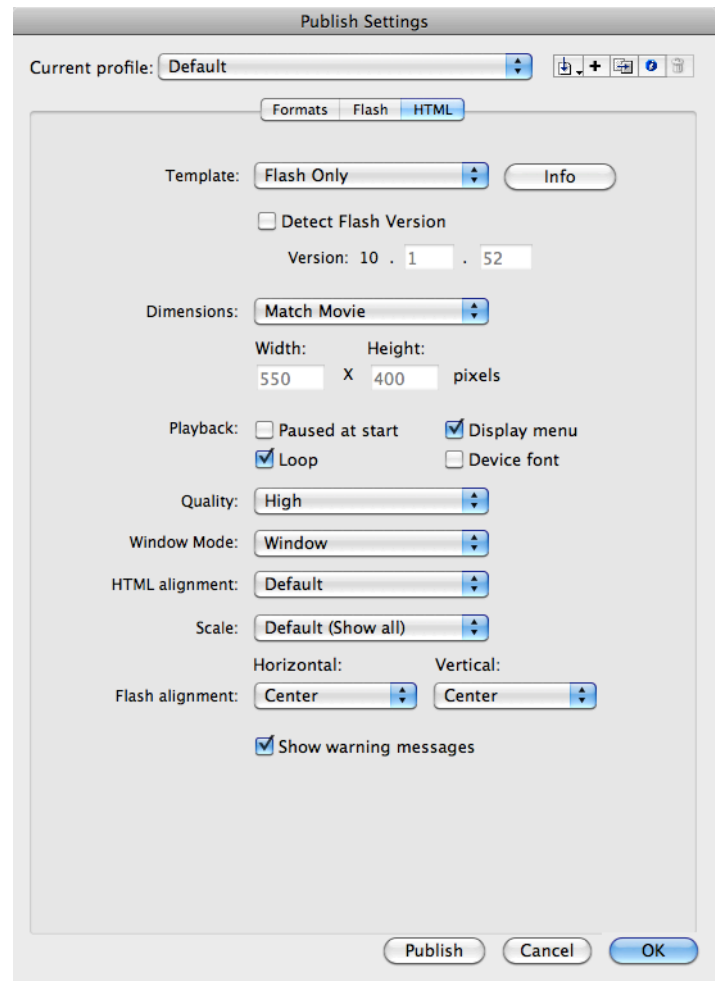
HTML and SWF

Most likely you will be sharing your Adobe Flash creations on a web site. Flash will publish out the HTML code in an .HTML file that you need to post your Flash file to the web.

Keep in mind that this code references your .SWF file so you will also need to upload that .SWF file to the web server along with the .HTML file/code.

Flash Player Versions

Another option of you get when publishing your Flash project is what version of the Flash Plug-in the file plays back in. There have been 10 main versions of the Flash Player since the beginning and your audience may have a wide variety of versions depending on how often they update their computer or how tech savvy they happen to be. This often is a guessing game and usually you will just leave the Flash Player version at the latest available. The Flash Player is a free plug-in for browsers available at Adobe's web site so your audience should easily be able to update their Flash Player version. If you are using the newer ActionScript 3.0 in your final .SWF Flash file you will need to be publishing out to at least the Flash Player 9.0.



Self-contained Projector files

If you don't want to distribute your Flash project over the web but rather share it with your audience to run on their desktop you can choose to create a Projector file for either Windows PC or Macintosh. Choosing one of these options will publish out a self-contained Flash file meaning that it will play back outside of the browser. This is useful for distributing your Flash projects on CD-ROM or emailing a project to a viewer. On the Windows PC platform Flash will export out an executable .EXE file. On a Macintosh the projector file Flash will export out is an application or .APP file type. These two types are files that the end-user simply double clicks and the Flash file will load and playback exactly as you see the file when you press Control and Enter to test your movie within Adobe Flash.

Most likely you will either use the HTML code and the .SWF file and push your Flash projects out to the end-user over the web browser or use a projector file for one of the two major operating system platforms (Windows PC or Macintosh).



Additional Resources

Flash CS5 and ActionScript 3.0 Books

ActionScript 3.0 Cookbook: Solutions for Flash Platform and Flex Application Developers by Joey Lott, Darron Schall, and Keith Peters (ISBN: 0596526954)

ActionScript 3.0 for Adobe Flash Professional CS5 (ISBN: 0321704479)

The ActionScript 3.0 Quick Reference Guide: For Developers and Designers Using Flash by David Stiller (ISBN: 0596517351)

Essential ActionScript 3.0 by Colin Moock (ISBN: 0596526946)

Foundation ActionScript 3.0 Animation: Making Things Move by Keith Peters (ISBN: 1590597915)

How to Cheat in Flash CS5 by Chris Georgenes (ISBN: 0240522079)

Learning ActionScript 3.0 by Rich Shupe and Zevan Rosser (ISBN: 059652787X)

Visual QuickStart Guide: Flash Professional CS5 Advanced by Russell Chun (ISBN: 0321720342)

Flash CS5 and ActionScript 3.0 Web sites

Lee Brimelow (Flash Evangelist for Adobe, Inc.)

[gotoandLearn.com](http://gotoandlearn.com) (tutorials)

FlashBlog.com (blog about Adobe Flash)

Chris Georgenes (Flash Designer and Author)

MudBubble.com (examples of Flash)

Keyframer.com (blog about Adobe Flash)

Variety of Web Resources

ActionScript.org (ActionScript tutorials)

<http://www.actionscript.org>

[CartoonSmart](http://CartoonSmart.com) (Flash and ActionScript tutorials)

<http://www.cartoonsmart.com>

[Flash My Mind](http://FlashMyMind.com) (ActionScript tutorials)

<http://tutorials.flashmymind.com/>

[Flash Speaks Actionscript](http://FlashSpeaksActionscript.com) (ActionScript tutorials)

<http://www.flashspeaksactionscript.com>