

CLASSROOM IN A BOOK® INSTRUCTOR NOTES

ADOBE® FLASH® CS3 PROFESSIONAL

Getting Started

The lessons in this book require assets, including Flash documents, that are included in the Lessons folder on the book's CD. Before beginning to work with the lessons, copy the files to each student's computer's hard disk.

Though Flash CS3 Professional requires a minimum of 512 MB RAM, Adobe strongly recommends using this book with a computer that has at least 1 GB RAM, especially when using larger video files, as in Lesson 9.

You may wish to use all the lessons in this book, or particular lessons. You can use the lessons in any order, but some tools and techniques are described in detail only in the first lesson in which they occur.

Introduce students to additional resources, such as Adobe Design Center and Adobe Developer Center, where they can access information provided by other Flash users and by Adobe.

Lesson 1: Getting Acquainted

Lesson 1 uses a sample project to provide an overview of the Flash workspace. Students are introduced to the Tools panel, Library panel, Property inspector, and Timeline. They also learn about layers and keyframes.

Starting Flash and opening a file

Demonstrate multiple ways of opening a document in Flash: double-clicking the Flash document in Explorer or Finder, starting Flash and choosing File > Open, and selecting the document from the Welcome screen that appears when you start Flash.

Emphasize the importance of saving the working copy of the file to the lesson's Start folder. In many lessons, Flash documents reference other asset files, which must be in the same folder as the Flash document.

Getting to know the work area

Identify and describe the following areas of the workspace: Timeline, Tools panel, Stage, Library panel, Property inspector, and the Edit bar above the Stage.

If you have Adobe Creative Suite 3 installed, you can show students the similarities between panels in Flash CS3 and panels in other Creative Suite 3 applications. The Timeline is similar to the Timeline in Adobe Premiere Pro or Adobe After Effects.

Understanding the Timeline

Invite students to move the playhead through (or scrub) the Timeline to see the animation. As you move the playhead, Flash displays the contents of the frame on the Stage. Hide and show different layers on frames to demonstrate how layers are stacked and that each frame may contain content from multiple layers.

Keyframes are essential in animation. A keyframe indicates a change, whether it be in the position or other qualities of something on the Stage or the beginning or end of an audio file. Students will use keyframes in every Flash project, so it's important that they understand the keyframe's purpose clearly.

Describe the difference between inserting a keyframe and inserting a blank keyframe. When you insert a keyframe, the keyframe contains the frame's original contents until you change it. When you insert a blank keyframe, all content is removed from the frame so that you can add new content.

Using the Property inspector

Demonstrate that the options in the Property inspector change to reflect what's selected. The Property inspector can display properties for the entire document, the selected frame, or the selected instance of a symbol. Which options are available for an instance depend on whether the symbol is a movie clip, graphic, or button symbol. (Symbols are covered in more detail in Lesson 3.) To apply values you've set in the Property inspector, press Enter or Return.

Using the Tools panel

The Tools panel contains several tools for selecting, drawing, editing, and navigating. Demonstrate that some tools are grouped together, and a tool may be hidden beneath another. To select a hidden tool, click the triangle in the icon of the displayed tool, and then select the hidden tool from the pop-up menu.

Beneath the tools is the tools options area. The options available depend on the selected tool. Demonstrate that the options area changes as you select different tools. When you select the Rectangle tool, the Object Drawing mode icon appears; when you select the Zoom tool, the Enlarge and Reduce icons are available; when you select the Pencil tool, the Pencil mode icons are available.

Lesson 2: Working with Graphics

Lesson 2 uses a sample project to introduce students to the drawing tools and simple animation. The project is an animation of a bubbling liquid to be used as a logo or as a preloader. As they create the preloader, students learn how to use the Rectangle, Oval, and Line tools; import a bitmap fill; make selections; group objects; and manipulate shapes. They also learn about the difference between drawing modes in Flash.

Flash drawing modes

By default, Flash uses the merge drawing mode when you draw rectangles, ovals, or lines. In merge drawing mode, the drawn elements are called shapes. Overlapping shapes merge together to form a single element on the Stage. When you select a shape, the selected area appears dotted. Though the elements merge together, you can select each separately. Each side of a rectangle's stroke is a separate element, for example. To select a shape's fill and stroke, drag the Selection tool around the entire shape.

In object drawing mode, active when the Object Drawing icon is selected in the options area of the Tools panel, rectangles, ovals, and lines are discrete elements, even when they overlap. The discrete elements are called drawing objects. When you select a drawing object, its fill and stroke are both selected.

In primitive drawing mode, active when you use the Rectangle Primitive or Oval Primitive drawing tools, drawn elements are discrete objects, but you can modify their corner angles or inner radius.

Merge drawing mode offers advantages in creating unusual, complex shapes. Object drawing mode lets you move objects without concern that they will change in relationship to other objects. Primitive drawing mode makes it possible to draw more complex shapes without having to merge multiple shapes together.

Invite students to draw and manipulate elements using each of these drawing modes so that they become familiar with the differences and can recognize them as they work with them. You can convert a shape to an object by selecting it and choosing Modify > Combine

Object > Union. You can convert an object to a shape by selecting it and choosing Modify > Break Apart.

Creating a simple animation

In the lesson, students use a simple motion tween to animate a bubble so that it moves from the bottom of the glass to the top, changing size as it does so. You can also use a Timeline effect to achieve the same result. Timeline effects provide simple animation options. To show students an alternative method of animation, demonstrate the Expand effect. (Choose Insert > Timeline Effects > Effects > Expand.)

The Expand effect expands, contracts, or expands and contracts objects over time. This effect works best with two or more objects grouped together or combined in a movie clip or graphic symbol. Objects containing text or letters work well with this effect. You can expand the duration in frames; expand objects, squeeze them, or both; expand direction; set fragment offset in pixels; shift the group center; or change fragment size.

Breaking objects apart

In this lesson, lines were created using the object drawing mode. To delete individual line segments, you need to break the lines apart, converting them into shapes. Emphasize for students the difference between shapes and objects in Flash, and the advantage of shapes when you want to select shape segments.

Masking objects

A Halloween mask hides what it covers; a graphic mask reveals what it covers. Students who have used Adobe Photoshop may already be familiar with masks.

In this lesson, the mask serves to keep the bubbles in the glass. The bubbles are revealed only where the mask exists, so they disappear when they reach the top of the glass. By default, a mask affects only a single layer beneath it, but in this lesson, students mask multiple layers. In later lessons, students will also animate masks to change the amount of an element that is revealed over time.

Lesson 3: Creating and Editing Symbols

Lesson 3 uses a sample project to introduce symbols, instances, and imported Illustrator files as students create a DVD interface. Students learn about buttons, rollover states, and transparency, as well.

Importing Illustrator files

Flash lets you import Illustrator AI files, and to a large extent preserves the artwork's editability and visual fidelity. The AI Importer also provides a great degree of control in determining how 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.

You specify how artwork is imported in the Import dialog box. You can select which layers—and which content on each layer—to import. Flash can import Illustrator layers as Flash layers, as a single layer, as keyframes, or as symbols.

In this lesson, students import an Illustrator file with layers that become Flash layers to make up the

background of the DVD interface. However, many of the same options are available when copying and pasting artwork. If Illustrator is installed on the computer, open the Illustrator file and copy and paste, or drag and drop, artwork from it to the Stage in Flash to demonstrate how the options are similar.

About symbols

A *symbol* is a graphic, button, or movie clip that you create once in Flash and then reuse throughout the document or in other documents. A symbol can include artwork imported 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 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 a document 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.

Movie clip symbols, button symbols, and graphic symbols each have benefits and limitations. Create one of each, and then show students the differences in the Timeline and in the Property inspector for each.

Button symbols automatically include rollover states in the Timeline. Demonstrate each rollover state: Up, Over, Down, and Hit.

- The Up state determines the button's default appearance when the mouse is nowhere near it.
- The Over state appears when the mouse rolls over the button.

- The Down state appears when the button is clicked.
- The Hit state defines the button's active area, which is the area that responds to the mouse rolling over or clicking the button.

Converting objects to symbols

In the Convert To Symbol dialog box, you name the symbol, select a symbol type, and specify a registration point.

A symbol's name, and especially an instance name, should always be lowercase or camelcase. In camelcase, a name begins with a lowercase letter, but each concatenated word is capitalized. For example, *pourBottle* is a symbol name in camelcase. Avoid capitalizing symbol and instance names because ActionScript reserves capital letters for specific classes and other keywords.

The registration point determines the point at which Flash reports the coordinates of the symbol, and from which it resizes or transforms the symbol.

Adding transparency

Transparency, or inversely, opacity, is determined by the alpha value for a symbol or a color. Decreasing the alpha value decreases the opacity and increases the transparency. You can change the alpha value for a symbol—which affects every instance of that symbol—or for a single instance.

Lesson 4: Working with Text

Lesson 4 uses an animated birthday card project to introduce students to creating, formatting, and animating text in Flash CS3. Students work further with keyframes, masks, and layers, and are introduced to filters.

Organizing layers in a Timeline

Layers are displayed on the Stage in the order in which they appear in the Timeline, with the bottom layer appearing first in the stack, the next layer up appearing on top of it, and so on, until the top layer in the Timeline is also the top layer on the Stage. As designers plan projects, they consider the order in which they want objects to appear. For complex projects, it's good practice to create layer folders and think through the order of the layers you want to use, rather than creating layers on the fly.

It is customary to create an Actions layer in Flash to contain all of the ActionScript for the Timeline, and to place the Actions layer at the top of the Timeline so that it is easy for the designer or others to find and access to troubleshoot or revise the ActionScript.

Adding a sound file

To add an audio file to a frame in Flash, first import it to the library. Then, select the frame at which you want the audio file to begin playing, and, in the Property inspector, choose the audio file from the Sound menu.

Selecting the appropriate option from the Sync pop-up menu is important. If a sound file doesn't play when you think it should, check the Sync menu to ensure you've selected the right option.

- **Event** synchronizes the sound to the occurrence of an event, such as a button click. An event sound plays when its first keyframe appears and plays in its entirety, even if the SWF file stops playing. If another event occurs, the first instance of the sound continues playing while the second instance of the sound plays.
- **Start** synchronizes the sound to the occurrence of an event, just as the Event option. However, 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 it can't draw animation frames quickly enough to keep up with the sound file, it skips frames. Stream sounds stop if the SWF file stops playing.

Next to the Sync menu is a pop-up menu that determines how many times the sound file plays.

- **Repeat** specifies the number of times the file plays; specify a number.
- **Loop** repeats the sound continuously.

Note: Repeating stream sounds is not recommended, because frames are added to the file and the file size is increased by the number of times the sound is repeated. You can also apply effects to the sound file.

- **None** applies no effect
- **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 Edit Envelope

Using tweens to animate objects

Motion tweens can animate the movement of elements on the Stage, as in the case of the bubble in Lesson 1, but they can also animate the change of objects' properties, such as brightness or transparency.

Point out to students that an arrow appears in the Timeline after you apply a tween to frames.

Creating text in Flash

Creating text in Flash is similar to creating text in Adobe InDesign, Adobe Photoshop, or numerous other applications. Use the Text tool to create an insertion point on the Stage and begin typing, or drag the Text tool to create a text box for your text. To format text, select it with the Text tool, and then select formatting options in the Property inspector. To manipulate a text box as an object, select it with the Selection tool and

make changes in the Property inspector or Transform panel, or simply drag the text box handles to resize it. Resizing the text box does not resize the text itself, but it does reflow the text in the text box. You can convert text to a symbol, and animate it similar to any other symbol. You can also apply a filter, such as a drop shadow or glow, to text.

In this lesson, students create static text, which appears exactly as it's typed on the Stage. In Lesson 8, students use dynamic text, which changes depending on viewer input or events in ActionScript.

Animating text with masks

Flash can recognize only one mask object on a mask layer. Therefore, to mask the text effectively in this lesson you must draw rectangles in merge drawing mode and be sure to overlap them, creating a single mask shape. For each change of the mask, students must first create a keyframe and then add to the mask.

Creating a blinking effect

When you insert a keyframe, the frame's contents are preserved until you change them. When you insert a blank keyframe, the frame's contents are erased. Therefore, you can use blank keyframes to create a blinking effect for the low battery symbol.

Adding keyframes to frames 206, 213, and 220 first ensures that each of those keyframes will contain the low battery symbol. Adding blank keyframes to frames 203, 210, 217, and 224 clears the content from those keyframes, so that the low battery symbol does not appear until the next keyframe. That is, the battery symbol appears between frames 206 and 210, between frames 213 and 217, and between frames 220 and 224.

Lesson 5: Adding Animation

In Lesson 5, students animate the title page for a space alien cartoon character as they learn about motion tweens, shape tweens, nested symbols, and motion animated along a path.

Working with nested symbols

Movie clip symbols have their own Timelines, similar to the main Timeline. A movie clip symbol may have other symbols nested within it. In fact, in this lesson, the alien symbol contains the alien_body symbol, which contains the alien_head symbol, which, in turn, contains the alien_head_eye symbol.

You can edit a symbol in place, in the context of other objects on the Stage, by double-clicking its instance on the Stage. You can edit a symbol in isolation, so that it is all that you see, by double-clicking the symbol in the Library panel.

To edit a nested symbol in place, double-click the symbol that contains it. If a symbol is nested within several other symbols, you need to navigate through all those symbols. When you're in a symbol's editing mode, Flash displays the symbol name in the Edit bar above the Stage. You can refer to the Edit bar to see which Timeline you're editing, and to see all the symbols it's nested in. To return to the main Timeline, click Scene 1 in the Edit bar. To go to any other symbol's Timeline in the path, click that symbol's name in the Edit bar.

Morphing objects

Shape tweens differ from motion tweens in that shape tweens actually morph one object into another. Flash interpolates the values or shapes for the frames between the first keyframe and the second keyframe, creating the animation.

You can apply shape tweening to drawn shapes. To apply shape tweening to groups, instances, or bitmap images, you must first break the elements apart. To apply shape tweening to text, as with the letter "O" in this lesson, break the text apart twice to convert the text to shapes.

Animating motion on 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.

For the best results, keep the motion guide smooth. Students may find it easier to draw a motion guide using a Wacom tablet or other pen pointer device, rather than a mouse.

Options in the Property inspector determine how the shape tween is created:

- **Orient To Path** - The baseline of the tweened element orients to the motion path.
- **Snap** - The registration point of the tweened element snaps to the motion path.
- **Sync** - The animation of graphic symbol instances is synchronized with the main Timeline. The number of frames in a tween is recalculated to match the number of frames allotted to it in the Timeline.

Lesson 6: Creating Interactive Files

Lesson 6 emphasizes interactivity, especially through the use of buttons and audio files. Students manipulate gradients, nest multiple kinds of symbols within a single symbol, and prepare a file for ActionScript. Lesson 7 introduces ActionScript fully, but the Lesson 6 file contains ActionScript that is summarized in a sidebar.

Creating buttons

The buttons in this lesson are button symbols that include nested movie clip and graphic symbols. Review the kinds of symbols and the benefits of each. Button symbols contain rollover states in their Timelines. Movie clip symbols can have drop shadows applied and include Timelines that can be animated just as the main Timeline can be. Graphic symbols let you repeat a single image multiple times without having to include the data multiple times. Additionally, you can create a graphic symbol that contains multiple images, and use a different image in each instance of the symbol.

Aligning buttons

The Align panel makes it easy to align or evenly space objects horizontally or vertically. You can align objects relative to the Stage or relative to each other. Demonstrate the options in the Align panel, and encourage students to experiment so that they are familiar with the different options.

Activating the buttons

The buttons' rollover states work without ActionScript, but ActionScript is required if the button clicks are to have any effect otherwise. Though students do not create ActionScript in this lesson, they do learn how to name instances and prepare a file for ActionScript. Emphasize the importance of naming instances so ActionScript can refer to them.

When naming instances, use lowercase or camelcase. It's often helpful to name an instance with an extension that identifies its symbol type. Flash designers typically name instances of movie clip symbols with an `_mc` extension, button symbols with a `_btn` extension, and graphic symbols with a `_gr` extension.

Name instances in a way that is meaningful, so you'll remember what you've called them when you're writing ActionScript. Shorter names are easier to type without error, but longer names may be easier to remember and to identify when you're editing the ActionScript later.

The ActionScript behind the scenes

Encourage students to view the ActionScript that is included in the lesson file. Select frame 9 on the Actions layer in the main Timeline, and then choose Window > Actions to open the Actions panel.

The ActionScript used in the lesson is described in the sidebar at the end of the lesson. Students will learn about ActionScript in Lesson 7 and can refer back to the ActionScript used in Lesson 6.

Lesson 7: ActionScript Basics

Lesson 7 introduces ActionScript 3.0, the scripting language that extends Flash functionality and enables interactivity in Flash projects. Students who haven't worked with scripting languages before may initially be intimidated and confused by ActionScript, which is an object-oriented scripting language similar to JavaScript. This lesson introduces basic scripting concepts, terminology, and syntax, as well as specific ActionScript coding.

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.

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 detailed information on ActionScript, see *Programming ActionScript 3.0*, *Learning ActionScript 2.0 in Adobe Flash*, or the *ActionScript Language References*, all available in Flash Help.

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.

Because there are multiple versions of ActionScript, and multiple ways of incorporating it into your Flash document files, there are several different ways to learn ActionScript.

Flash Help 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 available in Flash Help 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

Tutorial videos are also included on the Adobe Flash CS3 Professional Classroom in a Book CD. In particular, see the ScriptAssist.mov file from Lynda.com.

Adding a stop action

Students have had some experience with ActionScript and the Actions panel in earlier lessons because they've added stop actions. Stop actions are very simple; they stop the movement in the Timeline and prevent files from looping.

As you discuss stop actions, talk about the role of stop actions added in earlier lessons. Remove a stop action from the end file of an earlier lesson and preview the movie. Discuss what happens when the stop file is no longer there. In most cases, the movie or some part of the movie repeats.

Lesson 8: Using Components

Lesson 8 introduces students to components, which are pre-built movie clip symbols that you can customize for interactive user interface elements and video elements. In the lesson, students create two Text Area components, one with ActionScript and one without, and a Tile List component.

About components

You can create components in Flash by dragging them from the Components panel or by creating them in ActionScript. To add components using the Flash interface, drag them from the Components panel onto the Stage, set their parameters in the Parameters panel or Component inspector, and then assign behaviors and methods using ActionScript.

Note: You can use the Behaviors panel to assign behaviors to components if the document's ActionScript Publish setting is set to ActionScript 2.0. In this book, we always use ActionScript 3.0, so behaviors in the Behaviors panel are not available.

When you drag a component from the Components panel to the Stage, Flash imports an editable movie clip to the library. After a component has been imported to the library, you can drag instances of it to the Stage from either the Library panel or the Components panel.

You can set properties for each instance of a component in the Parameters tab of the Property inspector or in the Component inspector. You can also change the color and text formatting of a component by setting style properties for it or customize its appearance by editing the component's skins. Name component instances to refer to them in ActionScript.

Students can learn more about components by creating their own projects using components from the Components panel. Examples of components are included in the sidebar at the end of the lesson.

Lesson 9: Working with Sound and Video

In Lesson 9, students bring audio and video files together with ActionScript to create a dynamic, interactive wildlife preserve kiosk. As they work through the project, students learn how to edit audio files, import and optimize video files, and add ActionScript to load external files.

Editing a sound clip

In Flash, you can change the point at which a sound starts and stops playing, or control the volume of the sound as it plays. In this lesson, students remove half of the waveform of a sound to shorten the sound clip.

To edit a sound file, first add it to a frame, and click Edit in the Property inspector.

- 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.

Preparing Flash video content

Before you import video into Flash, consider what video quality you need, what video format to use, and how to download it. When you embed a video into a Flash document, it increases the size of the final SWF file that you publish. The embedded video starts progressively downloading to the viewer's computer whether or not they view the video.

You can also progressively download or stream the video at runtime from an external Flash Video (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, file sizes are smaller and load more quickly, and viewers download only what they want to see or use.

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. The FLV-Playback component in Flash CS3 Professional has a smaller file size than Media components in earlier versions of Flash. It is usually a good idea to give viewers some control (such as the ability to stop, pause, play, and resume the video, and control volume) over the video in a SWF file.

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.

Tips for creating Adobe Flash Video

Follow these guidelines to deliver the best possible Flash 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, 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 Device Central. If you need to reduce the frame rate, the best results come from dividing the frame rate by whole numbers.

Note: When you embed video clips in the SWF file, the frame rate of the video clip must be the same as the frame rate of the SWF file. To encode video using the frame rate of the FLA file, use the Advanced Video Encoding settings in the Flash Video Import wizard.

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 pre-

vent 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 Premiere Pro makes several Adobe Flash Video presets available through Adobe Media Encoder. 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.

Stream for best performance

To eliminate download time, provide deep interactivity and navigation capabilities, or monitor quality of service, stream Adobe Flash 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.

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: Pause = download time – play time + 10% 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 seconds – 30 seconds + 3 seconds = 33 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, all the Adobe Flash Video presets in the Adobe Media Encoder have deinterlacing turned on by default.

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.

Embedding video in a SWF file

Embedded video lets you embed a video file within a SWF file. When you import video in this way, the video is placed in the Timeline where you can see the individual video frames represented in the Timeline frames. An embedded video file becomes part of the Flash document.

When you create a SWF file with embedded video, 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. To use variable frame rates, import the video using either progressive download or Flash Media Server. When you import video files using either of these methods, the FLV files are self-contained and run at a frame rate separate from that of all other timeline frame rates included in the Flash SWF file.

You can import video clips into Flash as embedded files in QuickTime video (MOV), Audio Video Interleaved file (AVI), Motion Picture Experts Group file (MPEG), or other formats, depending on your system.

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.

Playing back external FLV 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 files in Flash Player. You can play FLV files posted as HTTP downloads or as local media files.

Create FLV files by importing video into the Flash authoring tool and exporting it as an FLV file. You can use the FLV Export plug-in to export FLV files from supported video-editing applications.

To play back an external FLV file, post an FLV 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 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 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 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 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 files can perform functions independently of the Flash document, and so do not interrupt playback.
- Captioning video content is easier with external FLV files because you can use callback functions to access metadata for the video.

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, you can export Flash 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 video as Flash content for further editing in Flash.

Exporting QuickTime video from Flash

If you create animations or applications with Flash, you can export them as QuickTime movies using the File > 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.

Exporting Flash video (FLV) from After Effects

When you render finished video from After Effects, select FLV as the output format in the render queue to export directly to the Flash Video (FLV) format. This launches the Flash Video Encoder application, which provides size, compression, and other output options. Any After Effects markers are added to the FLV file as cue points. You can then import the FLV 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 a FLV 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 file for composite effects.

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 runtime control of animation. After Effects is oriented towards 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 Flash's ActionScript language is the more robust of the two scripting environments. Both applications allow you to place graphics and effects on separate layers for compositing. These layers can be turned on and off as needed.

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 non-destructive 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 Flash project. When you export content from After Effects in SWF format, the content is flattened and rasterized in the SWF file.

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 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. When After Effects imports a SWF file, its internal keyframes are preserved so that you can continue to use them for timing other effects.

The Continuously Rasterized SWF Import feature in After Effects allows you to bring SWF file content into After Effects as flattened art, with support for alpha channels. Because the rasterization is continuous, vector art in the SWF file that is scaled scales smoothly in After Effects. 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.

Lesson 10: Loading and Optimizing Flash Content

Lesson 10 revisits the preloader graphic students created in Lesson 2 as students learn how to script the preloader and load a separate SWF file.

Working with preloaders

Preloaders entertain viewers and keep them informed while the primary content of a website is loading. Because website visitors are easily distracted and may click away from a site if nothing seems to be happening, many complex sites use preloaders. A preloader may be as simple as a progress bar, or it may be related to the content that is loading. Ask students to think of preloaders they've seen or preloaders they might use for different kinds of sites. (To find examples, try visiting the sites of graphic designers or sites that feature movie files.)

Previewing a preloader

A preloader only plays until the main content has loaded, so if you're running it on the computer that contains the content, the preloader will disappear quickly. To see what viewers will actually see, simulate a download.

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.

To test a movie:

1. Choose Control > Test Movie.
 2. In the preview window, choose View > Download Settings.
 3. Select a download speed to determine the streaming rate that Flash simulates. When viewing the SWF file, choose View > Bandwidth Profiler to see 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 Choose View > Simulate Download to turn streaming off or on. If you turn streaming on, the document starts over and simulates a web connection.

Optimizing 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.

- 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.
- 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.
- 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.
- 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.

Bitmap caching

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 un-stretched, 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.

Lesson 11: Publishing Flash Documents

In Lesson 11, students explore a range of options for publishing a document, using the alien animation they created in Lesson 5.

Publishing overview

By default, the Publish command creates a Flash SWF file, an HTML document that inserts your Flash content in a browser window, and a JavaScript file labeled `AC_RunActiveContent.js` that lets your SWF file play automatically in active content-compliant browsers. (For ActionScript 2.0, Flash creates a similar JavaScript file called `AC_OETags.js`.) 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.

You can publish the Flash document 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 Flash Player installed. When you publish a Flash document in alternative file formats, the settings for each file format are stored with the document.

You can export the Flash document 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.

Note: If you create a custom HTML document to display your SWF files in, ensure that it allows your content to play as expected in active content-compliant browsers. For more information about active content, see www.adobe.com/go/activecontent.

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 Multi-part 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.

Using Adobe Device Central with Flash

Device Central enables Flash users to preview how Flash files will look and function on a variety of mobile devices.

In the past, it was difficult for Adobe Flash Lite developers to test the files they created on mobile devices. Testing content could take a significant amount of time, especially manually exporting and testing on target devices and returning to Flash to make necessary changes.

Device Central is the next generation of mobile emulation and includes new features such as profile updates, memory and performance options, and custom device sets.

Device Central provides mobile content developers and testers with an easy way to create and preview mobile content on a variety of devices. Adobe Device Central displays realistic skins of a wide range of mobile devices that show you what the devices look like and how your content appears on those devices. This enables you to interact with the emulated devices in a way that simulates real-world interactions, including testing different performance levels, memory, battery power levels, and types of lighting.

Device Central provides a library of devices to choose from. Each device has a profile that contains information about the device, including the media and content types it supports (that is, the content that can be used on an individual device such as screen savers, wallpaper, and stand-alone Adobe Flash Player). You can search through available devices, compare multiple devices, and create custom sets of the devices you use most.

Device Central supports different media formats including Adobe Flash, bitmap, video, and web formats. You can use different media formats to create different types of content such as screen savers or wallpaper.

The Emulator tab in Device Central is designed to simulate content on mobile devices in a realistic way. You can test various media types such as Flash, bitmap, and video, and apply them as different content types, such as stand-alone player, wallpaper, or screen saver.

Note: Testing with the Emulator tab cuts the cost and time of testing on mobile devices, but should never replace testing on actual devices. Use Device Central for initial tests as you develop content and then use real devices for final testing.

If you are testing Flash content, for a content file to appear on the Emulator tab on a specific device, the device must support the Flash Lite version and content type that the file uses. For example, if you have a SWF file created in Flash that requires Flash Lite 2 and you

try to test the file on the Emulator tab on a device that only supports Flash Lite 1.1, the file does not appear. (In this case, try going to the Available Devices list, group the devices by Flash Lite version, and double-click one of the devices that supports Flash Lite 2.)

To test a Flash Lite document in a device, choose Control > Test Movie.