



SWF EXPORT FROM AFTER EFFECTS

"I'm not bad. I'm just drawn that way."

**—JESSICA RABBIT,
WHO FRAMED ROGER RABBIT**



REPURPOSING AFTER EFFECTS FILES FOR USE IN FLASH

One of the primary advantages of any digital medium is the potential to repurpose imagery or content from one means of delivery to another. In the world of design, this potentially means that a single visual language (logos, type treatments, print ads, and so on) can be developed once and then simply applied to a specific delivery method (printing, video, online, and so on). After Effects now has the capability to export SWF (Shockwave Flash, usually pronounced *swiff*) files, allowing motion-graphic artists to take their video or desktop multimedia work onto the World Wide Web.

This lesson will show how a short interstitial (used as a short introduction between episodes or commercial breaks) is translated from its original After Effects form into an optimized SWF and how this project can be optimized for web playback.

Project 3

SWF Export from After Effects

by Nathan Moody



How It Works

The reality of translating After Effects work to SWF files is never as quite as simple or as elegant as this “single-sourcing” theory sounds. For example, many broadcast designers create their work by starting with vector artwork, usually EPS files created from Illustrator or Freehand. This is also true of most Flash artists and animators as well. This at least provides original art assets that can be used for video, multimedia, and SWF media.

There are vast differences, however, between how QuickTime (or similar systems such as Windows Media Player) and Flash render their content. These differences can be summarized as follows: QuickTime and AVI movies have their pixels rendered before playback, whereas SWF files have their pixels rendered during playback.

After Effects normally renders its output as an absolute number of pixels: 720×486, 640×480, and so on. Each pixel is calculated in After Effects and rendered into a file, where each pixel is saved as part of a frame. QuickTime and AVI movies have fixed

resolutions, requiring a single viewing size that is optimal. The advantage of pre-rendered pixels is that complex effects, such as blurs, glows, and complex compositing, can be shown without taxing the computer’s processor because it has all been precalculated. As long as the device can keep up with the movie’s data rate (x number of pixels blasted to the screen y times per second), everything should play smoothly.

On the other hand, SWF movies represent shapes and forms as *vectors*, mathematically described angles and curves with certain color characteristics. SWF movies are rendered on the fly, during playback, by a SWF Player (Flash Player, QuickTime Player, and so on), or by a browser’s Flash plug-in on the viewer’s computer. Flash movies are resolution independent and can be dynamically scaled to accommodate any size of playback window. The advantage of SWF movies is that the files are miniscule compared to QuickTime or AVI files of comparable length, but SWF playback relies on the client/playback device’s computing power to

render its vector shapes. (RAM helps, but processor power is a much bigger factor in playback speed.) This is why SWF files remain one of the best ways to deliver rich content over low-bandwidth Internet connections.

Note: Mac OS users will find that SWF playback, in many cases, is quite slower than for their comrades on Windows machines. This is because the Flash plug-in for browsers and the Flash Player standalone application are not fully optimized for Mac OS playback. You'll see a significant increase in playback speed watching SWFs on the Windows platform. This is a critical tip in preparing media for the web: If your movie plays back really well on your Macintosh, it will *scream* when running in Windows.

GETTING STARTED

Before starting into the tutorial portion of this lesson, it is useful to take a look at After Effects' new SWF Export feature and discuss when and where this feature is best used.

After Effects' sophisticated numerical controls make it an excellent tool for creating complex, linear (noninteractive) SWF movies. Its capability to encode URLs into frames allows for simple, linear hyperlinking that will take the user to a different location after the movie is done playing. The main danger here is that many users are quickly tiring of the ubiquitous splash-page movies found on many SWF-enhanced web sites. Be sure what you create is short enough, small enough, and compelling enough to prevent your viewer from instinctively clicking that Skip Intro button.

Given that After Effects cannot embed hyperlinks in objects (although it can trigger URL redirects on certain frames, as you'll see later) or support interactive scripting, do not expect it to be able to export a functional animated interface. The main exception to this limitation is when small, intricate After Effects files are exported as SWFs and then imported back into Flash or LiveMotion as movie clips. This approach will let the artist build animations in After Effects' environment and do sophisticated scripting in a better tool: the best of both worlds.

When planning your After Effects project, consider whether it will need to be deployed online and build it appropriately. The SWF Export command is *not* well suited for putting your whole demo reel online. It *is* suited for evolving an ongoing campaign or corporate identity from print or video to the web.

INTRODUCING *THE GAMBLER*

The Gambler is an online animated series that profiles a randy "lover, not a fighter" antihero in 1970's Las Vegas. The producer of the show is unsure whether the series' niche and audience will be primarily online or on video, and he wants to hedge his bets by producing media that can be viewed both ways. Coupled with the lead animator's comfort level with After Effects as an animation tool, the show is made to be delivered both online and on video.

To this end, all the production design has been done in Illustrator: character design, backgrounds, everything. These assets are made in Illustrator and are rendered out to full-resolution DV QuickTime movies. However, they are also exported as SWF files for the show's web site.

Note: SWF files can be further edited and manipulated in either Adobe LiveMotion or Macromedia Flash. In fact, you'll be dipping into Flash later in this project to check your output results. If you don't have either program, it won't affect the outcome of this tutorial. However, if you do have either of these applications, I'd recommend running it alongside After Effects for this lesson.

LOOKING AT THE PROJECT: THE GOOD, THE BAD...

Given that this animation is inherently simple by design, you're not going to spend much time figuring out the specifics of how it was animated. You'll focus instead on how this composition is structured, and you'll analyze how well (or how poorly) it will translate into the SWF format.

Again, this piece was created specifically to be exported as both a DV-compressed QuickTime movie and as a SWF (complete with audio). To accommodate for Flash output, only two types of layers are being used: imported Adobe Illustrator artwork and After Effects solids. That's it—no pixels to be found.

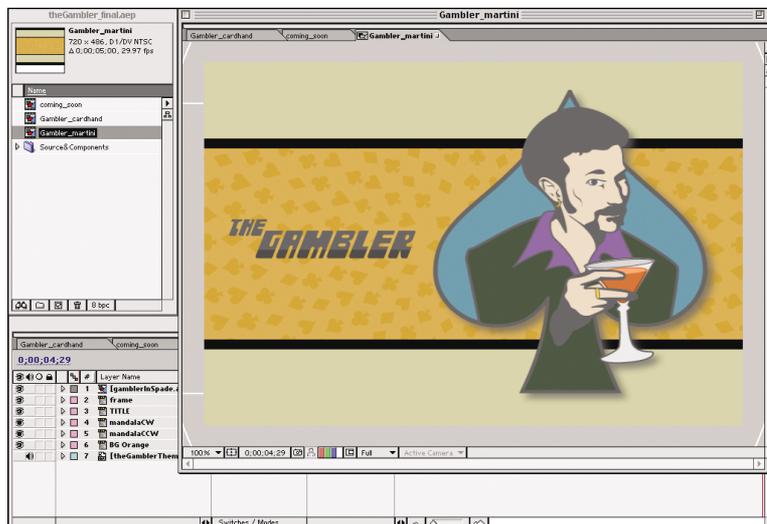
- 1 Open the **theGambler_final.aep** project file from this chapter's folder on the accompanying CD-ROM.
- 2 In the Project window, double-click on the composition called **Gambler_martini**; you'll see a finished animation ready for output.

The animation has a few layers moving around, and near the end of the piece, the gambler lifts his orange martini (hey, it's only a cartoon) as his logo comes into view.

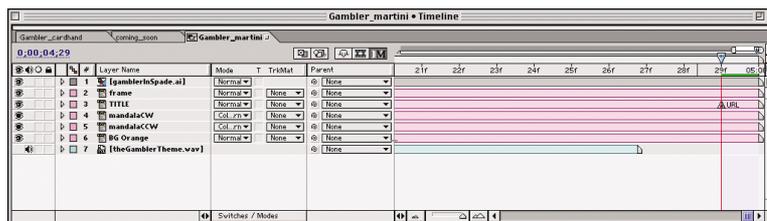
- 3 In the Timeline window, move your time indicator all the way to the last frame of the animation [End key]. Now, zoom in the Timeline's view as much as possible (by using the Zoom slider or by moving the left edge of the Time View bracket).

You'll see that there is a layer-time marker on Layer 3 (**Gambler_Title.ai**) at time 0:00:04:29, marked "URL."

Note: Take note of how the **mandalaCW** and **mandalaCCW** layers spin in opposite directions through the use of a simple expression. The pickwhip was used to link the two layers' rotation values together, and then the expression was multiplied by -1 to impart rotation in the opposite direction. This is another example of how animating with basic expressions can be *faster* than setting keyframes by hand.



This relatively simple, animated interstitial has been built completely with Illustrator shapes to facilitate SWF Export.



Layer-time markers can be used to produce simple URL links in exported SWF files.

- 4 Double-click on this layer-time marker to open its dialog box.

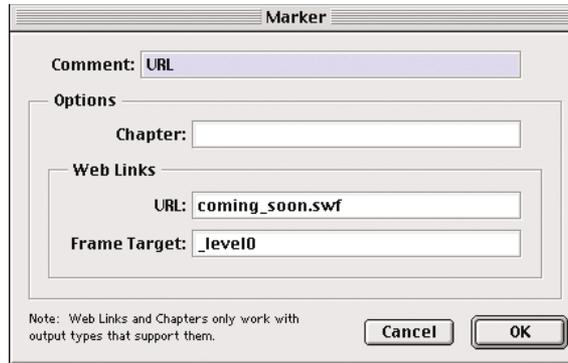
You'll see that it has a URL embedded in it; the last section of this lesson discusses its particulars.

Note: Chapter links, which can also be embedded in layer-time markers, can be used with QuickTime; they are most often used as a control for CD-ROMs and DVDs. They have no impact on SWF Export or viewing the movie in a browser. It's worth mentioning that composition-time markers have no linking functionality whatsoever; they are organization tools only.

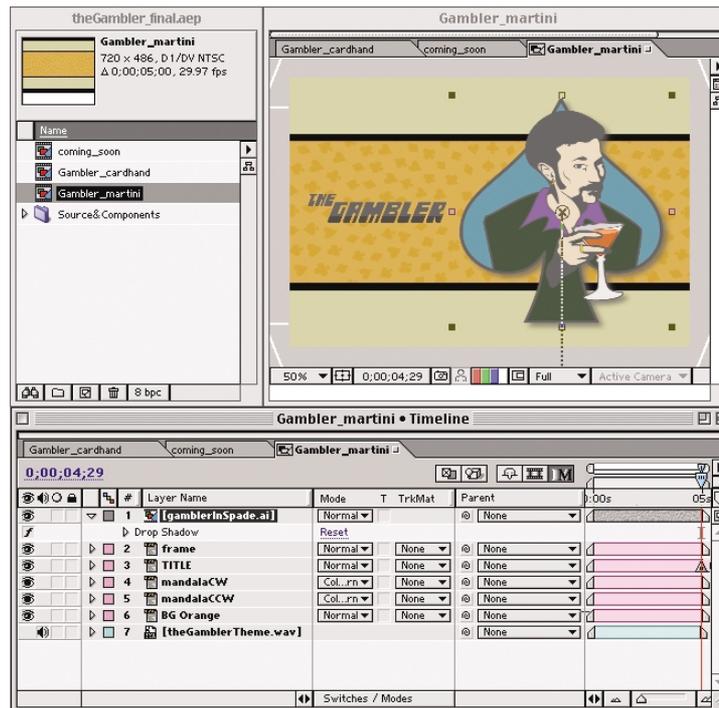
- 5 Zoom the Timeline all the way back out to the full length of the composition and select Layer 1 (gamblerInSpade.ai). Press the E key to inspect its effects.

This layer has a Drop Shadow effect applied to it. This is, in fact, the only effect applied in the whole composition. But let's take a moment to discuss why this minor effect is a potential problem.

SWF files easily accept vector or EPS artwork, but pixels are another story. This simple drop shadow not only generates pixels on its own, but these pixels are semi-opaque and are composited on a background with multiple solids. The SWF format is going to have a difficult time dealing with this because its format cannot assign transparency to individual pixels (although it can for vector shapes). Bitmapped graphics themselves can have an overall transparency or opacity value, but there is no accommodation for image-based masking or alpha channels in which color and brightness are figured on a per-pixel basis.



Time-layer markers support relative and absolute links as well as the capability to determine in which window or frame the link will be resolved.



The Drop Shadow effect is pixel based and can create problems when exporting vector-based SWFs.

So, you will probably want to lose the Drop Shadow effect for efficiency's sake, but yet another potential problem raises its ugly head: Motion Blur is also activated for the Gambler_Main composition. That is also a pixel-based effect. It might be time to examine some global ways of dealing with all these raster-heavy effects.

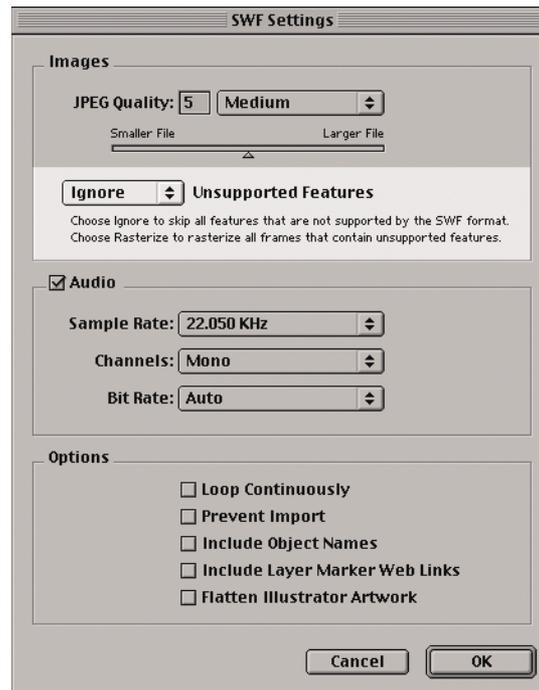
After Effects offers two solutions to these kinds of problems. One alternative is to simply ignore any setting or effect that will generate pixels, which is the route you'll take here (it will save on render time, anyway). Another solution is to rasterize all unsupported features as pixel-based imagery.

6 Select the SWF Export command (File > Export > Macromedia Flash [SWF]), name the file **gambler_martini_IGNORE.swf**, and save the file to somewhere on your hard drive.

7 Inspect the next dialog: SWF Settings.

You'll see that the second option down is the Unsupported Features preference, which can be set to either ignore or rasterize effects and features that generate pixels.

Don't close or resolve the dialog box yet.



The option to ignore or rasterize unsupported features is a major factor in determining how the SWF file will look... and how big it will be.

LOOKING AT THE PROJECT: ...AND THE UGLY

After Effects has an interesting habit that will make some animations very difficult to export: It always renders nested compositions as pixels. If you have any precomped layers or nested compositions in the composition you're trying to export, even if they are composed completely of vector or EPS shapes, you'll essentially have a large raster image in each frame.

Let's take a shot at exporting this composition as it stands.

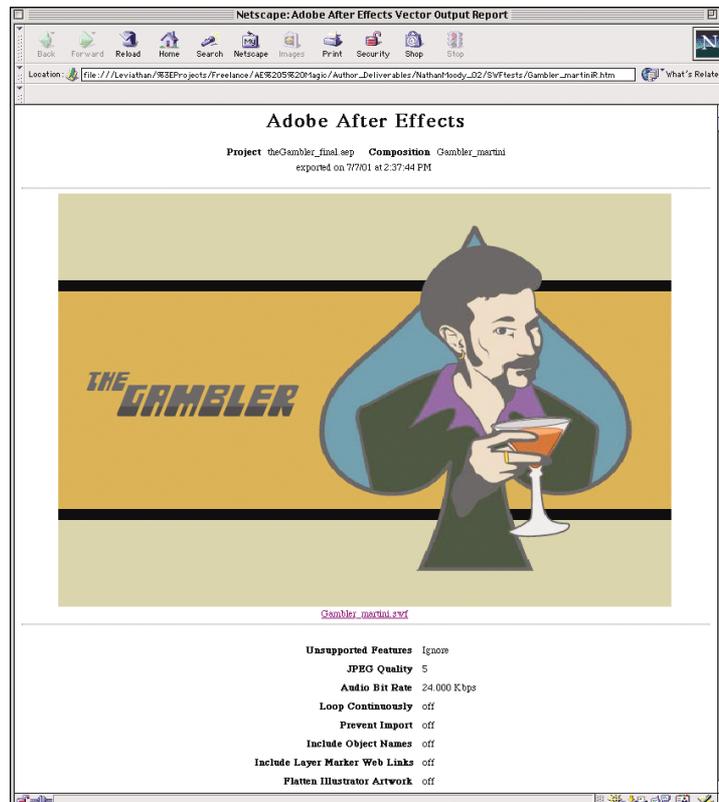
- 1 In the SWF Settings dialog box (which is hopefully still open from the last section), set the JPEG quality to 5 and ignore all unsupported features. Select 22.050 KHz audio and mono, with Bit Rate set to Auto, for the soundtrack. Leave all the other Options unchecked.

You'll notice that, although it might take a little while, this kind of rendering is much faster than actually rendering QuickTime movies. This is because After Effects is simply referring to the original Illustrator vector shapes and is reproducing those colors, points, and curves in the SWF format.

You'll also notice that two separate files have been created. `Gambler_martini.swf` is the actual SWF movie file, and `Gambler_martiniR.html` is an HTML document that enables you to view the SWF movie in a web page.

- 2 Double-click on the `Gambler_martiniR.html` file; it should open in a browser. (Drop-launch the file onto your preferred browser if this does not work.)

Note: MacOS users will find that the HTML files created by After Effects will be associated with the default browser specified in the Internet Control Panel. To alter this setting, go to the Apple menu and choose Control Panels > Internet. Click on the Web tab, and you will see the Default Browser pop-up menu at the bottom of the dialog box.



After Effects conveniently exports an HTML document showing the SWF and all of your export settings.

You'll see the animation play in a web page, but more interesting is all of the great information After Effects exports with each SWF render: all of your preference and file settings, render completion time, and even the name of the project and composition. Clicking on the hyperlink below the SWF will open it in its own browser window, if desired. This is very useful, especially for showing intermediate work to supervisors or sharing your work with other artists.

The best part about this web page, however, is its list of the unsupported features that the plug-in encountered, on a per-frame basis.

3 Scroll down the page, and you'll see a very long table with all sorts of error messages.

This is extremely helpful for troubleshooting After Effects projects and seeing how certain effects are (or aren't) handled.

Now for the bad news.

Note: If you want to see the animation play again—it currently is not set to loop continuously—press your browser's Reload button.

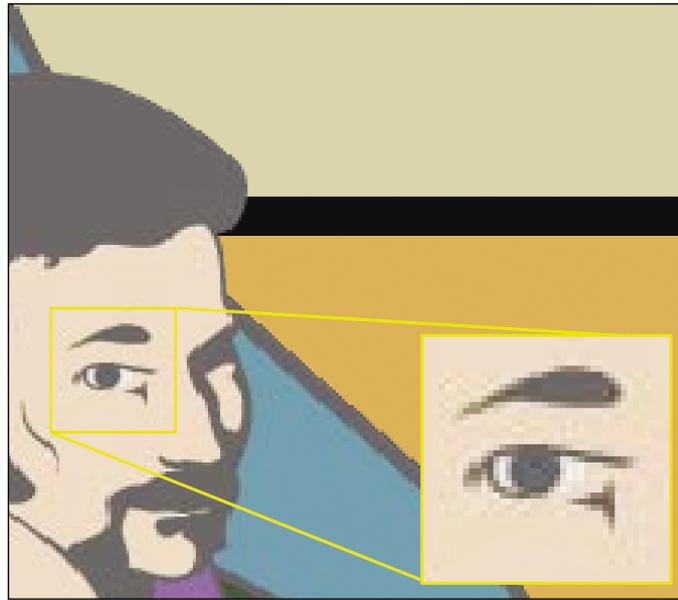
After Effects goes to great lengths to detail exactly which unsupported features were encountered during export.

Time	Name - Unsupported Feature	Unsupported Feature
0:00:00:00	1: gamblerInSpade.ai - multiple or unsupported effects 2: frame - motion blur 3: TITLE - motion blur 4: card_suits.ai - unsupported layer mode 5: card_suits.ai - unsupported layer mode 6: BG Orange - motion blur	1: gamblerInSpade.ai - multiple or unsupported effects 2: frame - motion blur 3: TITLE - motion blur 4: card_suits.ai - unsupported layer mode 5: card_suits.ai - unsupported layer mode 6: BG Orange - motion blur
0:00:00:01	1: gamblerInSpade.ai - multiple or unsupported effects 2: frame - motion blur 3: TITLE - motion blur 4: card_suits.ai - unsupported layer mode 5: card_suits.ai - unsupported layer mode 6: BG Orange - motion blur	1: gamblerInSpade.ai - multiple or unsupported effects 2: frame - motion blur 3: TITLE - motion blur 4: card_suits.ai - unsupported layer mode 5: card_suits.ai - unsupported layer mode 6: BG Orange - motion blur
0:00:00:02	1: gamblerInSpade.ai - multiple or unsupported effects 2: frame - motion blur 3: TITLE - motion blur 4: card_suits.ai - unsupported layer mode 5: card_suits.ai - unsupported layer mode 6: BG Orange - motion blur	1: gamblerInSpade.ai - multiple or unsupported effects 2: frame - motion blur 3: TITLE - motion blur 4: card_suits.ai - unsupported layer mode 5: card_suits.ai - unsupported layer mode 6: BG Orange - motion blur
0:00:00:03	1: gamblerInSpade.ai - multiple or unsupported effects 2: frame - motion blur 3: TITLE - motion blur 4: card_suits.ai - unsupported layer mode 5: card_suits.ai - unsupported layer mode 6: BG Orange - motion blur	1: gamblerInSpade.ai - multiple or unsupported effects 2: frame - motion blur 3: TITLE - motion blur 4: card_suits.ai - unsupported layer mode 5: card_suits.ai - unsupported layer mode 6: BG Orange - motion blur
0:00:00:04	1: gamblerInSpade.ai - multiple or unsupported effects 2: frame - motion blur 3: TITLE - motion blur 4: card_suits.ai - unsupported layer mode 5: card_suits.ai - unsupported layer mode 6: BG Orange - motion blur	1: gamblerInSpade.ai - multiple or unsupported effects 2: frame - motion blur 3: TITLE - motion blur 4: card_suits.ai - unsupported layer mode 5: card_suits.ai - unsupported layer mode 6: BG Orange - motion blur
0:00:00:05	1: gamblerInSpade.ai - multiple or unsupported effects 2: frame - motion blur 3: TITLE - motion blur 4: card_suits.ai - unsupported layer mode 5: card_suits.ai - unsupported layer mode 6: BG Orange - motion blur	1: gamblerInSpade.ai - multiple or unsupported effects 2: frame - motion blur 3: TITLE - motion blur 4: card_suits.ai - unsupported layer mode 5: card_suits.ai - unsupported layer mode 6: BG Orange - motion blur
0:00:00:06	1: gamblerInSpade.ai - multiple or unsupported effects 2: frame - motion blur 3: TITLE - motion blur 4: card_suits.ai - unsupported layer mode 5: card_suits.ai - unsupported layer mode 6: BG Orange - motion blur	1: gamblerInSpade.ai - multiple or unsupported effects 2: frame - motion blur 3: TITLE - motion blur 4: card_suits.ai - unsupported layer mode 5: card_suits.ai - unsupported layer mode 6: BG Orange - motion blur
0:00:00:07	1: gamblerInSpade.ai - multiple or unsupported effects 2: frame - motion blur 3: TITLE - motion blur 4: card_suits.ai - unsupported layer mode 5: card_suits.ai - unsupported layer mode 6: BG Orange - motion blur	1: gamblerInSpade.ai - multiple or unsupported effects 2: frame - motion blur 3: TITLE - motion blur 4: card_suits.ai - unsupported layer mode 5: card_suits.ai - unsupported layer mode 6: BG Orange - motion blur
0:00:00:08	1: gamblerInSpade.ai - multiple or unsupported effects 2: frame - motion blur 3: TITLE - motion blur 4: card_suits.ai - unsupported layer mode 5: card_suits.ai - unsupported layer mode 6: BG Orange - motion blur	1: gamblerInSpade.ai - multiple or unsupported effects 2: frame - motion blur 3: TITLE - motion blur 4: card_suits.ai - unsupported layer mode 5: card_suits.ai - unsupported layer mode 6: BG Orange - motion blur
0:00:00:09	1: gamblerInSpade.ai - multiple or unsupported effects 2: frame - motion blur 3: TITLE - motion blur 4: card_suits.ai - unsupported layer mode 5: card_suits.ai - unsupported layer mode 6: BG Orange - motion blur	1: gamblerInSpade.ai - multiple or unsupported effects 2: frame - motion blur 3: TITLE - motion blur 4: card_suits.ai - unsupported layer mode 5: card_suits.ai - unsupported layer mode 6: BG Orange - motion blur
	1: gamblerInSpade.ai - multiple or unsupported effects	1: gamblerInSpade.ai - multiple or unsupported effects

- 4 Look at the file size of your SWF: over 900KB, almost 1MB! Second, take a very close look at the SWF's upper-right corner and take account of what's happened.

In After Effects, the spiraling mandala of card suits in the background is a reddish-orange color, thanks to the Color Burn mode. Flash and the SWF format have no idea what blending modes are, so it has completely ignored these layers (as per our instructions).

There are some blocky artifacts around edge regions of the gambler himself but nowhere else. These are JPEG compression artifacts, evidence of the aforementioned tendency to rasterize nested compositions. The emblem with the character in it had to be precomped to allow for a drop shadow on the entire image.



Even though all elements were created in Illustrator, some shapes become rasterized—and compressed as JPEGs.

There is another quite subtle issue with this SWF file: its pixel aspect ratio. Because it was created with DV in mind, its pixel aspect ratio is not square, making it look wider than it should on a computer monitor. Take a look at the following figure to compare how NTSC and DV pixel aspects differ.

If you needed to re-render a QuickTime movie for a different pixel aspect ratio, that's nothing a little pre-compositing can't cure. However, as you've already learned, precomps become rasterized, so that's not a viable alternative. Each frame would be a different JPEG, at which point it's better delivered as a QuickTime or AVI movie.

The two quickest workarounds for this problem lie outside of After Effects. One solution is to scale the SWF in the browser by simply entering new width parameters in a web page's OBJECT or EMBED tags. Alternatively, in Flash, all its frames can be



A subtle but noticeable difference can be seen between square pixels for standard NTSC video and nonsquare pixels in digital D1 and DV footage. SWF Export does not correct for rectangular pixel aspect ratios.

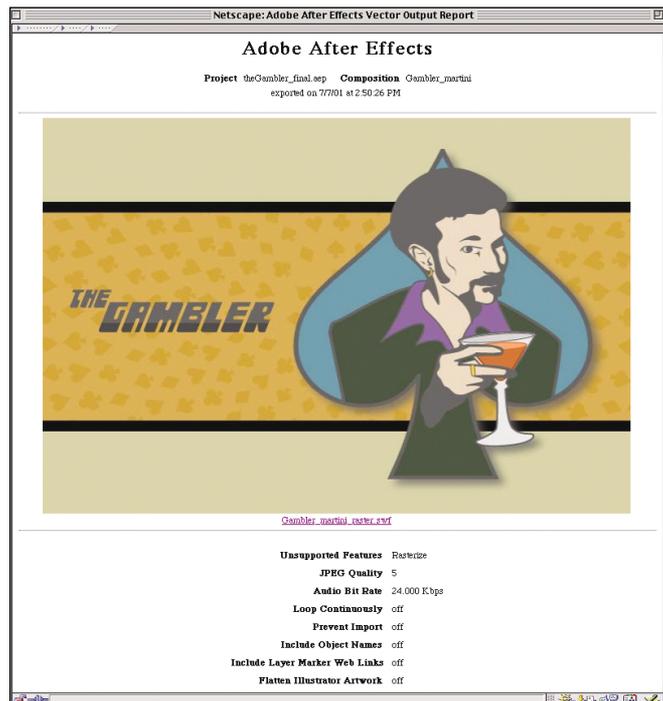
imported (or cut and pasted) into a movie clip, and then its instance can be scaled after it is placed in a Scene. Because SWFs are vector-based, they hold up very well to resizing like this, although text-heavy SWFs might become hard to read.

A final workflow suggestion is to create such clips with square aspect ratios from the start and then precomp them into a new composition with a DV-aspect pixel ratio for high-resolution output. This enables your SWF files to come out better, but it's not of much use for altering preexisting After Effects projects and compositions.

- 5 Back in After Effects, re-export the Gambler_martini composition and name the SWF **Gambler_martini_raster**. When the SWF Settings dialog box appears, choose Rasterize for the Unsupported Features option.
- 6 Open the resulting HTML file in a browser.

Now you have your drop shadow and your mandala back. However, look at the file size: over 3.5MB! What happened? Between your precomp, the mandala layers' unsupported color mode, and the drop shadow, every single frame has been saved as a JPEG. Not exactly what the SWF file format was meant to deliver over the web, is it?

All in all, this composition (and its resulting SWF) is quite a mess. Now, you haven't optimized any of After Effects' output settings, which could help a bit, but even the 900KB+ original file size is not good for a 5-second online clip. It might be time to move on to greener pastures.



Rasterizing the unsupported features brings back every detail from the original After Effects file.

REINTRODUCING *THE GAMBLER*

The preceding example showed some of the pitfalls that artists and animators will encounter with the SWF Export feature in After Effects. Retrofitting compositions made for other media (specifically DV) can have some serious limitations.

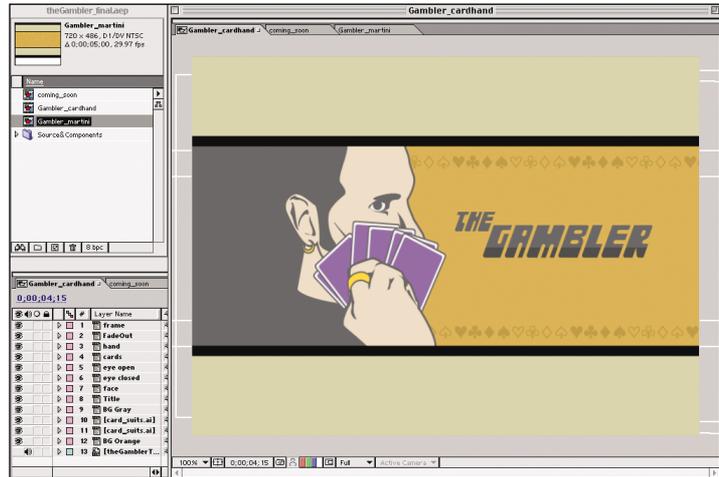
Luckily, there is an alternate composition in this project with which you will have much better luck: Gambler_cardhand. Luckily, it was also built as a short interstitial, and it will fit the same role for the show as the previous composition.

- 1 Set the Composition window's resolution to Half and view it at 50%. Then create a RAM preview [0 on the numeric keypad].

This animation is a simple move of the gambler's profile from left to right, with an occasional blink and a hand of cards moving over his lower face.

There is not a drop shadow in sight, no motion blur to be found, and no troublesome precomps. Even all of the blending modes have been set to Normal, and the pixel aspect ratio of this composition is square. This clip is a good example of a composition that has been built for SWF export first and DV output second. Altering the composition to a nonsquare pixel ratio can be accomplished by nesting it in a DV-aspect composition (and scaling the precomp accordingly).

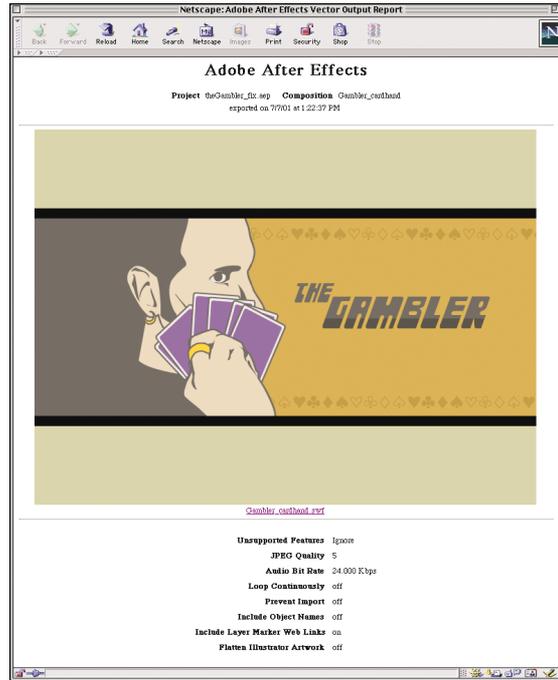
- 2 Select File > Export > Macromedia Flash (SWF) and save the resulting SWF to your hard drive.
- 3 In the SWF Settings dialog box, reset the Unsupported Features option back to Ignore. There should be no surprises at output time.



This second composition is cleaner, simpler, and lacks any pixel-based effects and nested compositions.

- 4 Open the HTML file that After Effects has produced in a browser.

The first thing you'll notice is that the SWF looks exactly like the original and plays back at a decent speed. If you scroll to the bottom the HTML file, the table of unsupported features is missing: Everything you've done in the After Effects project has been translated perfectly into the SWF format. Most importantly, the file size of this animation is less than 60KB, a dramatic improvement over your last attempt with the previous composition.



This SWF is completely vector based. There are no horizontal scroll bars in this browser window, which means there are no unsupported features to report.

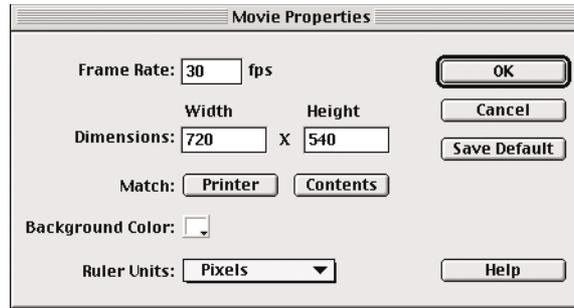
UP CLOSE AND PERSONAL: EXAMINING SWF OUTPUT IN FLASH

The SWF file you just created is ready to go on a web site. You can even copy and paste the EMBED and OBJECT tags right out of the Gambler_cardhandR.htm file into a new HTML page. However, many animators will want to import this SWF into Flash or LiveMotion to be used in a larger interactive piece. Let's examine how After Effects' SWF output imports into these applications, and let's see what you can learn about optimizing your SWF movies.

If you have Flash or LiveMotion, this is a good time to launch the application and follow along, but these steps will be "shorthanded" for those who do not have access to one of these tools. This lesson will use Flash.

First, the Movie Properties of a Flash movie must be set to whatever the After Effects project dictates.

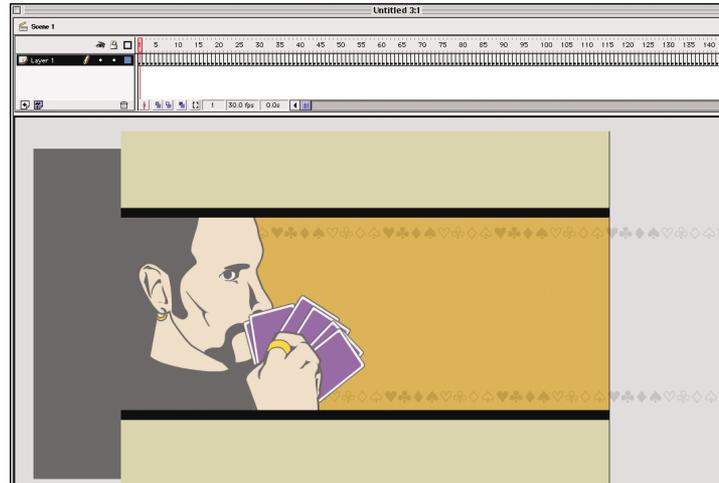
- 1 Before importing the SWF, your Flash movie should be set to 30 frames per second, and its dimensions should be 720 pixels wide by 540 pixels tall.



In Flash, these settings will accommodate your exported SWF for further study and editing.

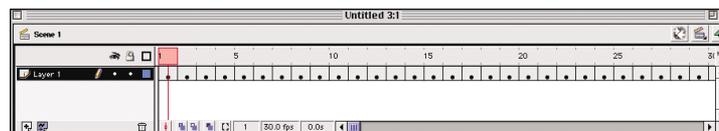
- 2 Import the SWF into Flash.

At first glance, there are no big surprises. The clip is indeed 150 frames long. (Remember that Flash starts frame numbering at 1, not zero.) There are some shapes beyond the stage area, but that's to be expected because you're moving around some objects that are bigger than the active stage.



Your exported SWF, in Flash, shows the full extent of all shapes on the stage.

One thing that will immediately be a concern to some Flash artists is that all the components of the animation are imported on a single layer, and *every* frame is a keyframe. This means that customization or refinement of motion or movement must be done in After Effects because the imported SWF contains only frame and image information, not tweening.

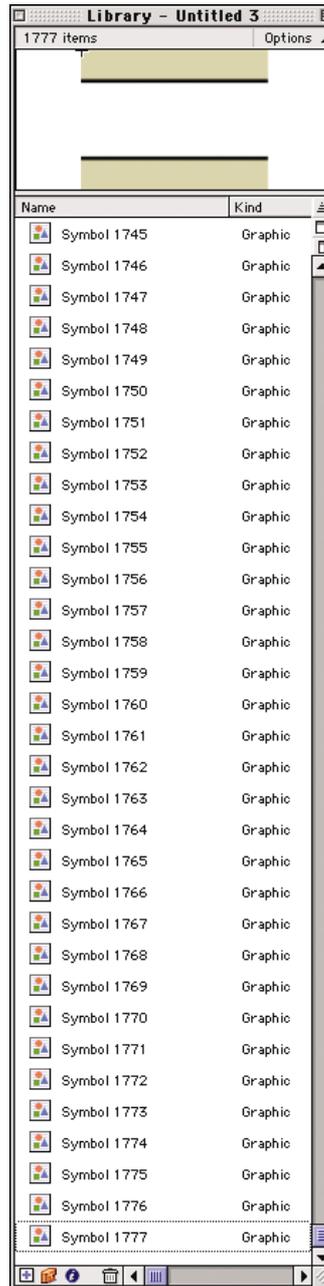


Notice that every frame contains a keyframe, and all the frames are in a single layer.

3 Open the Library to see the components that this SWF brought with it. The Library contains 1,777 symbols!

What's even more interesting is that not every symbol is different. The first 130 to 140 symbols are unique: they comprise every line, arc, stroke, and fill contained in the imported artwork. You'll also find a few large red boxes in the Library; they were not imported into your original After Effects project, but the SWF uses them as masks to accomplish certain shape compositing tasks. At a certain point (after Symbol 140 or so), however, every 22nd symbol seems to be the same: For example, Symbol 1733, Symbol 1755, and Symbol 1777 are all the beige "letterboxing" bars on the top and bottom of our animation. If you save this Flash project, you'll see that all of this winds up yielding a FLA file well over 3MB in size.

Note: The most unfortunate result of importing this SWF into Flash is that its audio does not get imported into Flash. If you are building a larger Flash movie and want to present this SWF with audio, the best way to accomplish this is by using the Load Movie ActionScript command, which will load and play external SWFs into another SWF at runtime.



Every object on every frame gets its own symbol. This exported SWF, brought into Flash, has more than 1,700 symbols.

FINISHING THE SWF ANIMATION

So, having looked in-depth at your gambler SWF in Flash, it's time to get back into After Effects and complete your animation. You'll take the interactivity that After Effects offers you and link a separate SWF into your main Gambler_cardhand composition.

Note: Loading external SWF files into other SWFs using the GetMovie ActionScript feature is an effective way to create modular content, which can be edited separately and modified according to project and client needs.

- 1 In After Effects, open the Gambler_cardhand composition and double-click on the layer-time marker at frame 0:00:04:29 in the Title layer.

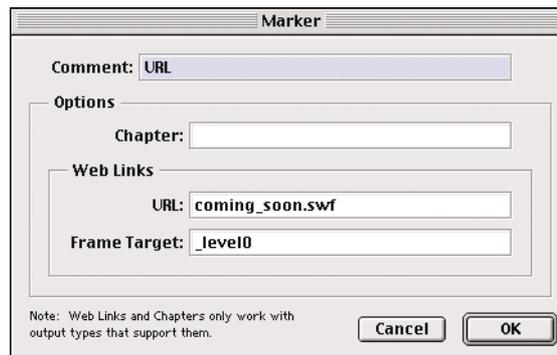
This URL is a relative link to another SWF file in the same directory. The Frame Target, `_level0`, refers to the SWF's main Timeline. When this animation reaches this frame, this marker will trigger an ActionScript command to automatically load a separate SWF into this movie.

- 2 Click OK to close this dialog box.

- 3 In the Project window, double-click on the `coming_soon` composition.

This composition is an extremely simple title card that states the release date of the full animated series.

As you can guess from this composition's name, this is the SWF that will get loaded into the Gambler_cardhand SWF. For this trick to work, you'll have to export each of them separately and put them both into the same directory.



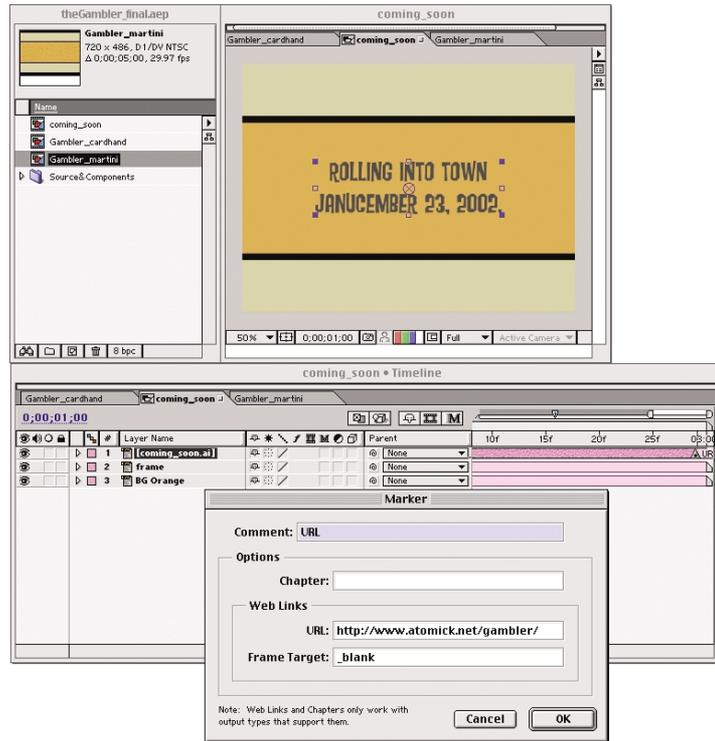
This layer-time marker's Web Links settings will translate into an ActionScript command that will load a new SWF right into the current one.

Note: For those After Effects artists who are unfamiliar with Flash (or who are intimidated by its scripting capabilities), there is good news. Both After Effects' Expressions and Flash's ActionScripting language are built on JavaScript-style syntax. With some minor adjustments for each application, an artist or programmer only needs to learn one type of syntax to be able to program effectively in JavaScript, ActionScript, and Expressions. Purchasing a single book on basic JavaScript is a great investment for any artist working in dynamic digital media.

- In the Timeline window for the coming_soon composition, there is a layer-time marker at time 0:00:02:29 in the coming_soon.ai layer. Double-click on this marker to open its dialog box.

This marker is a little different than the previous one: It is loading a true URL for a web page, not a SWF file. Its Frame Target is set to `_blank`, which will open a new browser window with the listed URL.

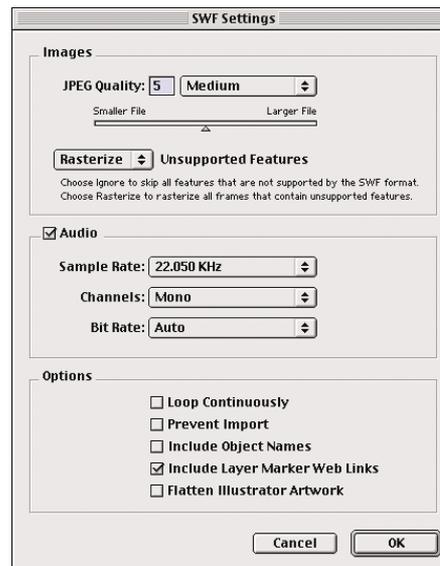
Note: Sometimes a layer-time marker's GetMovie action gets triggered before the image content of the frame loads. To prevent a single-frame "ghosted" text (the last frame of the Opacity tween before the coming_soon.ai layer becomes transparent), make sure all keyframes are set one frame before the layer-time marker.



This type of web link will resolve as a standard URL.

- Export the Gambler_cardhand composition and make sure to check the Include Layer Marker Web Links option.
- Do the same to the coming_soon composition. Because the coming_soon.swf file points at a real URL, enable an Internet connection.
- Open the Gambler_cardhandR.htm file in a browser. If you have Flash, simply double-clicking the Gambler_cardhand.swf file will open it in Flash Player, which will still respect URL hyperlinks.

You'll see the gambler animation play first, then the "coming soon" animation will load in its place, and finally a live web page will be loaded from over the Internet.



Include Layer Marker Web Links must be checked to enable hyperlinking and ActionScript commands.

FINAL ANALYSIS: WORKFLOW FOR SWF EXPORT

- It is almost always better to build After Effects compositions *specifically* for SWF deployment. It is easier to manipulate such compositions to conform to video specifications than the other way around (although it can be done).
- In building projects that will be brought into Flash or LiveMotion, it's most helpful to still consider After Effects a specialized animation tool and Flash or LiveMotion an integration and editing environment. It is most efficient to only animate what is actually moving in After Effects and then mix those components with static elements imported or copied directly from Adobe Illustrator or Macromedia Freehand. Likewise, final audio tracks need to be imported into Flash, although working with them in After Effects will help achieve synch.
- If you want to play an After Effects–exported SWF within a larger Flash movie, it makes more sense to do so using the Load Movie or GetMovie ActionScript commands. This will avoid the loss of audio that will occur when a SWF file is imported into Flash. You can even link several individual SWF files together in After Effects itself through the use of layer-time markers.
- Leaving the Include Object Names and Flatten Illustrator Artwork options unchecked in the SWF Settings dialog will usually yield smaller files.
- Creating complex motion is what After Effects is all about, so use it for what it is good for. Simple animations might be better done entirely within Flash or LiveMotion if those packages are available. This mainly pertains to file size; SWFs exported from After Effects are not completely optimized for file size.

MODIFICATIONS

You should continue to experiment with this project and get a handle on the SWF Export feature's strengths and weaknesses. The gambler's lovely purple 1966 GTO is also included in the After Effects project (in the project's Footage folder) for your further animation edification, as are all of the original Illustrator files.



The gambler's regal wheels are included for further animation edification.