

Online Theatre: Making Software Demonstration Movies

BY STEVEN BROOKS
Senior Member, Santa Barbara Chapter

You've probably seen them: online movies that demonstrate software in action. From marketing demonstrations to online training, there is a trend toward using moving pictures to communicate about software.

Such movies are not solely the purview of marketing and training. Software documentation can also take advantage of demonstration movies. Where you might expect to find a still screen capture, instead picture an online movie—a moving screen capture. Using movies can increase audience understanding, ease anxiety, and even energize users.

For example, the online help for Macromedia *Flash* integrates movies to illustrate its animation features (see Figure 1). In this case, demonstration movies go beyond slick eye candy to become a rapid and compelling communication tool.

It's time technical communicators took advantage of demonstration movies. But where do you start? This article offers an overview of the key decisions you must make when "breaking into movies."

Using Movies—When and Why?

Demonstration movies are appropriate for documenting not only graphics software, but any software. Here are some sample documentation components:

- Overviews involving time or workflow
- Tutorials and getting-started demonstrations
- Illustrations of moving graphics, such as animation and data plotting
- Procedures with abundant windows, menus, and dialog boxes

The benefits of movies are clear once you watch some examples (see Table 1 for a list of Web sites). Compared with text, moving demonstrations are easier to absorb and understand, especially because they directly mirror actual software behavior. They're a refreshing alternative to text for users faced with the anxiety of learning software and following complex tasks.

Are demonstration movies always the right medium? Absolutely not. Our audience needs text for quick access, or-

ganization of ideas, and printing. Also, because of the cost and effort required to create online movies, management might be opposed. However, creating movies is easier and cheaper than ever before, putting them within the reach of any writer and almost any documentation budget. Once managers see a working example, it's fairly easy to sell them on the use of movies in documentation.

Choosing a Movie File Format

One of the first choices you must face in creating an online movie is the file format. Although there are a wide variety of file formats for online movies, the following are the most popular choices:

- Macromedia's *Flash*
- Apple's *Quicktime*
- Microsoft's *Windows Media*

All of these formats require associated players, but these players are nearly ubiquitous on all operating systems and Web browsers. There are certainly other formats, but for software demonstration movies, the above three are predominant.

Let's start with *Flash*. On the Web, *Flash* movies have appeared in abundance for silly animations like those on jibjab.com. But the same technology is well suited for demonstration movies. *Flash* is vector-based, meaning that it can transfer images as geometric descriptions and "redraw" the images in your *Flash* player. That technology makes *Flash* movies small in file size, clear in quality, and able to be animated and made interactive. The interactive features make *Flash* ideal for e-learning, because you can prompt your audience for input during the movie.

The other two formats, *Quicktime* and *Windows Media*, are based on streaming media technology, so called because the player receives media in a data stream and plays it as it arrives. Unlike the vector images in *Flash*, streaming media uses bitmapped images, making them ideal for showing fluid motion and live action video. This fact can mean a larger file size, but by adjusting the video compression, you can trim the size. Although *Quicktime* and *Windows Media* are available on all operating systems, you should verify that the required players are readily available for your audience. For example, some Unix systems may need special configuration to play these file types.

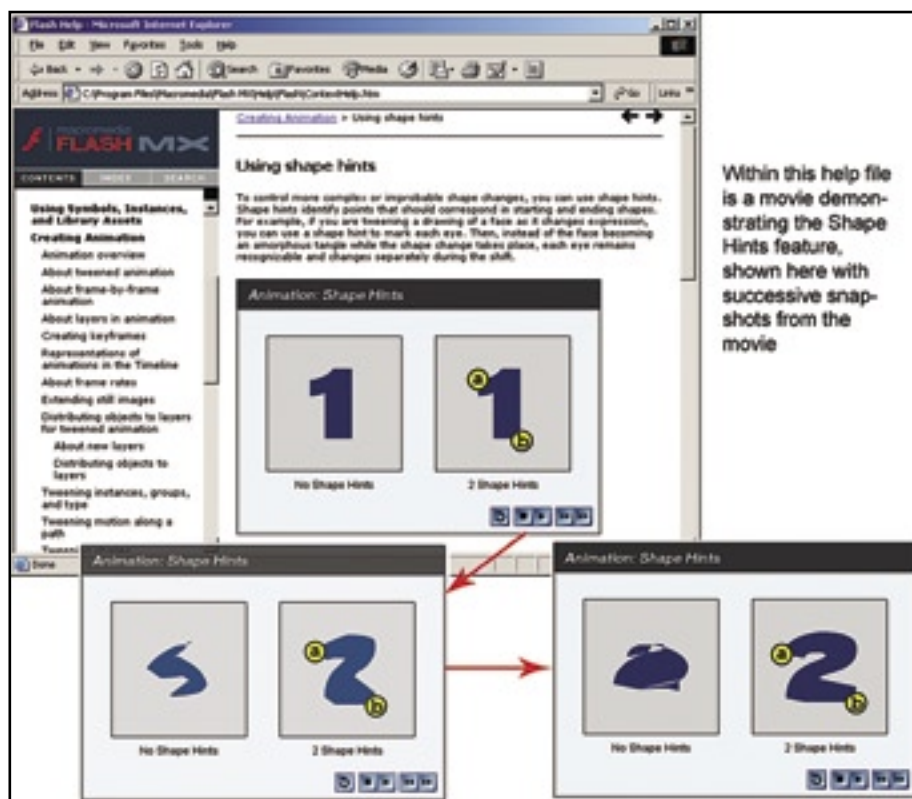
Your choice of format does not have to be absolute. It's also possible to insert streaming media clips into a *Flash* project. However, your final decision banks largely on the recording tool you choose and the format it supports.

Choosing a Recording Tool

The choices for screen recording tools are abundant and come in a variety of price ranges. Most tools run only on Windows, but you can also find tools for capturing on Macintosh and Unix. Table 2 lists a few of the common software titles.

For the most part, all of these tools provide the same core features. First, they record your screen activity: mouse movements, menu choices, typing, and so on. They also let you edit that recording by changing the timing, integrating audio, authoring text callouts, and, in some cases, adding interactivity.

Figure 1. This *Flash* help file contains a movie demonstrating the Shape Hints feature, shown here with successive snapshots from the movie.



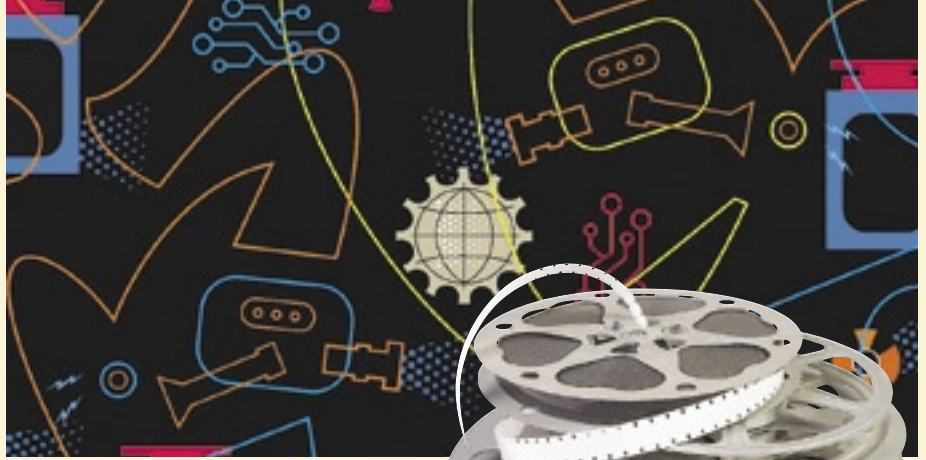
Within this help file is a movie demonstrating the Shape Hints feature, shown here with successive snapshots from the movie

Table 1. Sample Demonstration Movies

File Format	Web Location
<i>Flash</i>	www.macromedia.com/software/captivate
<i>Windows Media</i>	office.microsoft.com/en-us/FX010804491033.aspx
<i>Quicktime</i>	www.lynda.com

Table 2. Common Screen Recording Tools

Product	Company	Captures on	Output Format
<i>Captivate</i>	Macromedia	Windows	<i>Flash</i>
<i>Camtasia</i>	TechSmith	Windows	<i>Flash</i> or various streaming media
<i>Snappz Pro</i>	Ambrosia Software	Macintosh	<i>Quicktime</i>
<i>ViewletBuilder</i>	Qarbon	Windows or Linux	<i>Flash</i>



**You can make
movies to fit
any budget, any
help system, and
any skill level.**

Increasingly, the ability to add interactivity is one of the most powerful features and is the reason many of these recording tools output to *Flash*. For example, you can quiz your audience or prompt them to perform steps in your simulated interface. In fact, an interactive movie is typically called a *simulation* rather than a *demonstration*. For e-learning in particular, look for a tool that has robust features for creating interactivity.

One caveat for *Flash* is that you must limit the overall number of images captured from screen. The more images in the *Flash* animation, the larger the file size and the slower it opens. For this reason, tools like *Captivate* and *ViewletBuilder* capture only a small number of screen images and combine them with an animated replica of the mouse movement.

In general, this approach works for most screen recording needs. However, for more dynamic and fluid screen activity, such as scaling an image, a better choice would be a screen recorder like *Camtasia* or *Snapz Pro*. These programs capture complete screen activity, mouse movements and all. Their primary file output is streaming media, like *Quicktime*. However, it's possible to embed a streaming media clip into a *Flash* file; in fact, *Camtasia* also outputs to *Flash* format.

Choosing Between Audio and Text Callouts

Should you make a "talkie" or a silent movie? Audio does give added impact and ease of use to movies. However, it is not necessary and may be an annoyance in your customers' work environment.

The alternative to audio is a series of text messages that are timed to appear throughout playback: in other words, an animated slide show. You'll also find that many of the screen recording tools help to automate the creation of callouts. Macromedia's *Captivate* even writes the callouts for you, based on the software menus and buttons you choose.

From a writer's perspective, there are some advantages to using text callouts: smaller movie file size, easier editing, and easier translation. However, working with audio can be relatively easy too. Your choice depends largely on what medium is best for your audience.

Whether you use audio or text, your comments should be brief and well timed. In a movie, you can be brutally sparse in your descriptions, relying instead on the screen movement and visual highlights to make your point. For example, you don't need textual clues about where something is in the interface; you just highlight that area momentarily.

Timing, as they say in comedy, is everything. In most cases, give the comment, and then demonstrate it in the action. For example, explain the menu item the user needs to choose, and then show that menu selection being made.

Integrating Your Movie with Online Documentation

The best way to integrate movies into documentation depends on your needs. First, consider whether the movie file is kept locally on the customer's machine or on your company Internet site. For the convenience of viewing movies while offline, install locally. For disk space savings during installation, keep movies on an Internet site.

Second, consider whether to embed the movie within the online text, like an image, or to launch it in a separate player window. If you embed it within text, you need to keep the width and height of the movie to a minimum. If the software you're documenting takes

a large amount of screen real estate, you'll need to employ tricks such as increasing your screen resolution, turning off unnecessary parts of your software interface, and recording just a portion of the software.

If you launch movies separately, you need to design the navigation method between them. The navigation method could range from links in a help page to menu items on your software help menu. Typically, streaming media movies (*Quicktime* and *Windows Media*) work best if launched in a separate player window, which ensures the best quality and reliability.

Whatever your integration method, the skills required are minimal. Most of the screen recording tools provide ways to publish your movie into an HTML file. Also, the major help authoring applications have features for embedding movies into compiled help systems like HTML Help.

In general, you can make movies to fit any budget, any help system, and any skill level. When you do, you'll find that not only the audience enjoys documentation more, but the author does, as well. ❶

Steve Brooks is a technical writer and trainer for DreamWorks Animation in California. His prior experience includes writing for various software companies. He can be reached at stevebrooks3@earthlink.net.