Foundation Flash 8

Sham Bhangal and Kristian Besley

Foundation Flash 8

ISBN (pbk): 1-59059-542-4

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com.

For information on translations, please contact Apress directly at 2560 Ninth Street, Suite 219, Berkeley, CA 94710. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit www.apress.com.

The source code for this book is freely available to readers at www.friendsofed.com in the Downloads section.

Credits

Lead Editor
Chris Mills

Technical Reviewer
Todd Yard

Editorial Board
Steve Anglin, Dan Appleman, Ewan Buckingham, Gary Cornell, Tony Davis, Jason Gilmore, Jonathan Hassell, Chris Mills, Dominic Shakeshaft, Jim Sumser

Project Manager
Beth Christmas

Copy Edit Manager
Nicole LeClerc

Copy Editors
Damon Larson and Liz Welch

Assistant Production Director
Kari Brooks-Copony

Production Editor
Linda Marousek

Compositor and Artist
Katy Freer

Proofreader
Patrick Vincent

Indexer
Present Day Indexing

Cover Image Designer
Corné van Dooren

Interior and Cover Designer
Kurt Krames

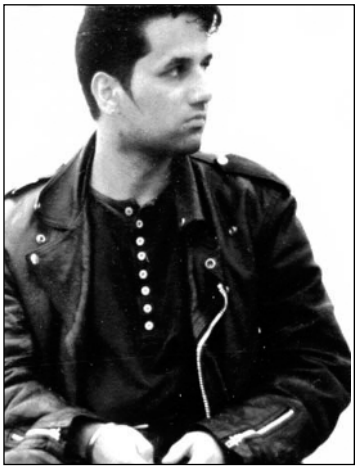
Manufacturing Director
Tom Debolski

Case study illustrations courtesy of Gareth Southwell at www.woodpig.com

CONTENTS AT A GLANCE

CONTENTS

ABOUT THE AUTHORS



Sham Bhangal began on the route to web design in 1991, designing and specifying information screens for safety critical computer systems, as used in places like nuclear power plant control rooms. He soon discovered that more conventional interface design, animation, and multimedia tools were available, such as SoftimageXSI, Photoshop, and Flash. He has been writing books and articles on them since the turn of the century.

As well as being an author by night, Sham works as a Flash developer for an e-learning company during the day, creating online testing applications, simulations, and other web-based teaching applications using lots of class-based ActionScript and a fair bit of user interface and content design. He is also known to engage in copious amounts of freelance web design work.

Like all professional web designers, he never sleeps because new ideas and old clients have a habit of keeping him awake. You can catch up with him at `boy@futuremedia.org.uk`, or through his weblog, `http://weblog.motion-graphics.org/`.

Portrait image courtesy of Simon James at `www.thefresh.co.uk`.

Kristian Besley is a Flash/web developer currently working in an educational establishment and specializing in interactivity and dynamic-driven content (using ASP and PHP). He is also a lecturer in multimedia.

He has written a number of books, including *Flash MX Video*, *Learn Programming with Flash MX*, and the Foundation Flash series. He was a contributor to *Flash Math Creativity*, *Flash MX Games Most Wanted*, and *Flash Video Creativity*. He is also a contributor to *Computer Arts* magazine and has produced freelance work for numerous clients, including the BBC.

Kristian was born and currently resides in Swansea, Wales. He is a fluent Welsh speaker and is the creator of the first ever Welsh translation search plug-in for Firefox and Mozilla (available from `http://mycroft.mozdev.org`).

ABOUT THE TECHNICAL REVIEWER

Todd Yard is a lead Flash developer at Brightcove (www.brightcove.com) in Cambridge, Massachusetts, and has contributed as an author or technical editor on 11 other friends of ED Flash books, including *Flash 8 Essentials* and *Extending Flash MX 2004*.

ABOUT THE COVER IMAGE DESIGNER

Corné van Dooren designed the front cover image for this book. Having been given a brief by friends of ED to create a new design for the Foundation series, he was inspired to create this new setup combining technology and organic forms.

With a colorful background as an avid cartoonist, Corné discovered the infinite world of multimedia at the age of 17—a journey of discovery that hasn't stopped since. His mantra has always been "The only limit to multimedia is the imagination," a mantra that is keeping him moving forward constantly.

After enjoying success after success over the past years—working for many international clients, as well as being featured in multimedia magazines, testing software, and working on many other friends of ED books—Corné decided it was time to take another step in his career by launching his own company, *Project 79*, in March 2005.

You can see more of his work and contact him through `www.cornevandooren.com` or `www.project79.com`.

If you like his work, be sure to check out his chapter in *New Masters of Photoshop: Volume 2*, also by friends of ED (ISBN: 1-59059-315-4).

INTRODUCTION

This book aims to give you a solid foundation in the most essential skills you need to use Flash 8—both the Basic and Professional versions. By the end of the book, you'll understand how the components of a Flash movie fit together, you'll have used all of the key tools, and you'll have integrated all your learning in a series of detailed creative exercises. Our mission is to launch you into orbit around planet Flash, equipped with all the tools and knowledge you need to make a safe landing.

Flash is one of the hottest content-creation technologies on the Web. From its origins as an animation package, Flash has grown stronger and planted deep roots. It is already used to create all kinds of content, such as website front-ends, interactive games, animated cartoons, movie trailers, and PDA interfaces. Perhaps its most significant role, however, is in creating *interfaces* for all these different areas. Its ability to present a clean, friendly, and functional front-end to the user is coupled with its power behind the scenes. Designers love Flash for its speed, quality, ease of use, and clearly structured functionality, and at the same time, both programmers and designers can use its ActionScript programming language to produce phenomenal results. Whatever kind of interface you want to build, Flash has the answer. If you've never used Flash before, you're in for a real treat.

As the Internet has changed, Flash has moved with it, evolving into a two-tier system. The timeline-based animation is still there, but it is underpinned by a stronger emphasis on functionality that enables you to create the large, code-heavy sites required for today's e-commerce front-ends, e-learning applications, and other intelligent user interfaces. Don't worry if you are not a heavy-duty programmer, though; you can still build Flash sites using many of the new features aimed directly at designers who are not programmers:

- If you are experienced in bitmap editing applications such as Adobe Photoshop, you will be able to leverage this experience in Flash, because Flash now supports many of the features seen in Photoshop, such as filters and blend modes.
- If you are more of an Illustrator/Freehand person, then you're also going to like Flash, because its new Object Drawing mode will make you feel right at home.
- If you are a video content producer, then Flash is also for you, because it is now one of the most powerful over-the-Web video-delivery systems.
- If you are a musician, then be prepared to learn a system that allows you to give your creations the web presence they deserve: 32-channel sound at the quality and format of your choosing!

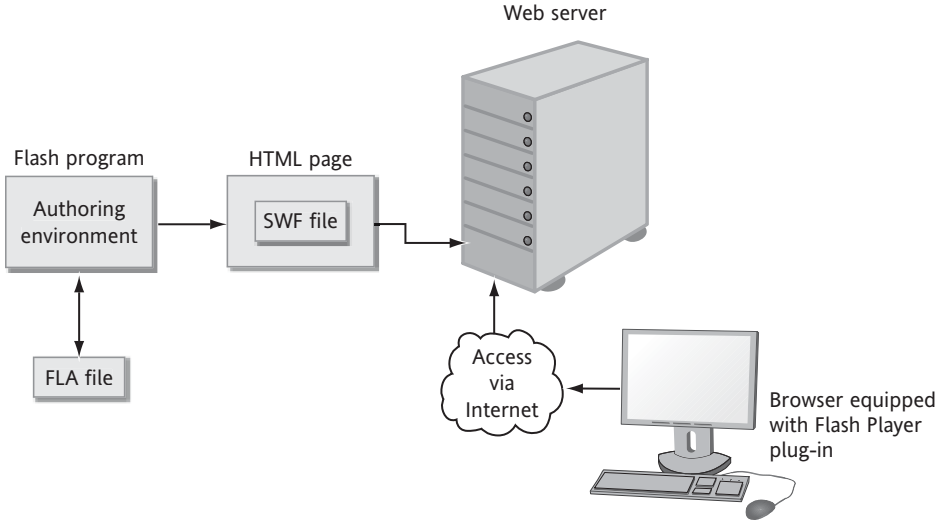
- Finally, if you are a traditional web designer, then you are in for a treat, because Flash already supports many of the things you are already familiar with (ActionScript, a JavaScript-like scripting language; support for subsets of both CSS and HTML; and runtime import of most of the common bitmap formats you are already used to). It also seamlessly addresses some of the things you have always had trouble with, such as consistent color, one single plug-in for all multimedia types (sound, video, animation), embedded fonts, multilanguage localization, and of course, a good interactive web animation system. Flash also addresses areas in which you may have previously thought Flash was weak, such as accessibility and security.

This book will take you step by step through every aspect of designing your own Flash interface, building your knowledge and skills with each chapter. We'll also look at the pitfalls and practicalities that every Flash web designer faces, teach you how to make your designs web friendly, and ensure that you know how to get your hard work up on the Net. But before we dive into these complex issues, let's make sure you know the basics of how Flash works, and why it's such a capable authoring tool.

Flash—the big picture

When you create a Flash movie for the Web, you're pulling together images, sound, video, text, and animation, and bundling them in a file that gets posted up on a website.

The Flash software you install on your computer is the authoring environment in which you create your masterpiece. The work in progress is stored in a file with the extension `.fla`. Once you're happy with your movie and you want to publish it to the Internet, Flash will convert the FLA file into a playable file with the extension `.swf`—pronounced "swiff" in the Flash community. The SWF file is then linked to an HTML file on the server that hosts your site:



When a user visits your site, the SWF file is downloaded into their browser and your movie is played back. All the viewer needs is the Flash Player installed on his machine. This player is a reasonably compact download, and the vast majority of the world's browsers are equipped to play back Flash content.

One of the reasons that Flash is such a popular tool is that it uses **vector graphics** technology. There are two main graphic standards on the Internet: **raster** (bitmaps) and **vector**. The majority of static images that you see on the Web are raster images, composed of files in formats such as BMP, GIF, and JPG. Raster images do a good job, but a big raster image usually requires a large file size, and a large file size means a long download time. And on the Web, download time is *everything*. Internet users are picky—if a site's packed with raster images and is taking too long to load, they'll just skip it and go somewhere else. This is where vectors come in. They're small, fast, and funky.

> *Vector images describe the image in terms of coordinates and mathematical transformations. That sounds complicated, but it's really as simple as saying, "Put a dot here, put a dot there, and draw a line between them." This compares with the raster technique of describing the color and position of every single pixel in the image.*

Vector graphic files are much more compact and efficient compared to rasters, and Flash is the main tool for delivering vector graphics and vector-based animations on the Web. The files that Flash creates are therefore comparatively small, which is one of the reasons for Flash's success.

A well-constructed Flash file will also **stream** onto the user's computer. That means it will load the first part of the animation and start playing it back while the rest of the animation loads in the background. Streaming a file correctly is an important technique for a Flash designer because it means that visitors are presented with something visual and enticing almost immediately—removing the danger that they will get bored and go elsewhere instead of waiting for the site to download.

Another disadvantage of raster images is that they're *display dependent*, meaning that if you create them to look just right on one particular display, the image could come out significantly altered if someone uses a different display resolution to view it. In addition, if you *zoom in* on a raster image, the pixels just get bigger and bigger until you end up with a screen full of squares of color that are completely unrecognizable as the source image. Vector images, though, can work independently of the display because the line will always be the same relative length and clarity no matter what resolution you use to view it. Also, no matter how far you zoom into a vector, the image will still stay crisp and at full resolution.

Why would you ever want to use a raster? Raster formats are good for images with thousands of different colors. Can you imagine trying to describe a photograph in terms of vectors? It would be horribly complicated, and you'd wind up with a far bigger file size compared to the raster equivalent. Luckily, Flash has the best of both worlds: the vast majority of its drawings and animations are vector-based, but when you need the extra richness that you can only get with a raster, Flash will allow you to import a bitmap and use it in conjunction with the dominant vectors.

Despite the fact that Flash is normally associated with vectors, Flash 8 now also includes many features normally associated with bitmaps, and this feature makes Flash much more accessible to people who are coming to Flash from more traditional digital art disciplines (especially graphic design and traditional web design).

What's significant about Flash 8

If you've used a previous version of Flash, the first thing you'll notice is. . .er. . .that it appears like nothing much has changed! Looks can be deceptive, though, because a whole lot has changed—so much so that many well-known and well-respected Flash designers are already calling Flash 8 *the* most significant Flash release in years!

Most of the new features do not involve the look and feel of the interface, but the things you can do with it.

For the designer or digital artist looking to create a web presence with Flash, there is a whole lot of new stuff. First, you now have lots of little tweaks to the tools and general interface that make it all easier to work with. Changes to gradient production, tweens (custom easing), the way vectors are drawn (Object Drawing), and tabbed panels are some examples of this.

Second, there are totally new features such as filters and other effects that will revolutionize your workflow. Stuff you would normally have had to do in a separate application, or do manually (such as adding drop shadows or bevels to buttons, or creating logos) can be done quickly and easily within Flash. Better still, doing them in Flash results in effects that are very bandwidth friendly. Further, all the new visual effects can be animated, allowing you to create animated content that was just not possible with previous versions of Flash. If you've ever used Photoshop layer blend and filter effects, you are in for a happy surprise—Flash 8 supports them, and—get this—you can animate these effects in real time! Not only can you apply these effects to bitmaps and vectors (including vector animations), but you can also apply most of them to video as well!

Third, there is now an updated scripting help system, called Script Assist. This helps nonprogrammers to write ActionScript, Flash's scripting language. ActionScript is the doorway to many of Flash's high-end visual effects, as well as the thing that allows you to add that all-important ingredient, interactivity, to your Flash content. Script Assist is one way of getting through that doorway with the minimum of fuss (and more importantly, without a lot of programming knowledge!).

One of the most significant subsets of the new enhancements for designers are those enhancements that are within the Flash Player itself.

The Flash 8 Player has several optimizations that allow you to create animations and user interfaces that run much faster than previous Flash content. This allows you to create more complex effects, bigger and better sites, or just simply smoother, more fluid animations. The Flash 8 Player now lets you import PNG files at runtime, allowing you to import images with embedded alpha channels on the fly. The Flash 8 Player also has much improved video facilities, a better sound engine, and a better text rendering engine that results in clearer, sharper text.

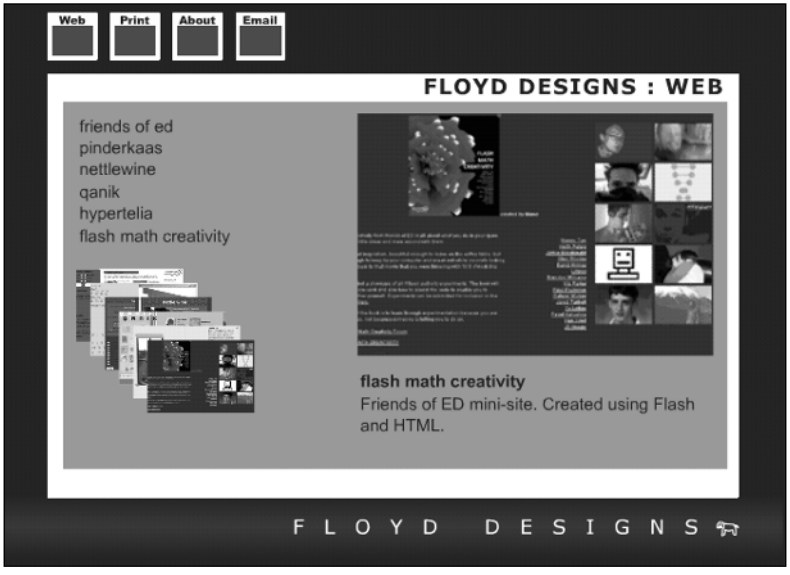
Of course, because the new features in Flash 8 are not obvious, it makes having a guide to help you along all the more important. That's where this book comes in!

Our aims and philosophy in this book

As its title suggests, the aim of this book is to give you a solid, extensible foundation in Flash design, implementation, and programming. We believe that Flash is too complex a tool to cover definitively from scratch in 1,000 pages, let alone 600-odd. We want to provide a rock-steady foundation: an in-depth treatment of the core aspects of using Flash rather than an overview of each and every feature.

We believe in creating a reliable foundation so you can understand Flash more fully and absorb and internalize the material we cover. We're not going to list every menu option and cover every single ActionScript command in immense detail. We're going to concentrate on the core of learning Flash successfully, taking you from a zero knowledge of Flash to being able to put up a website you can be proud of.

Everyone knows that the best way to learn is to play and practice. It's no good if someone just *tells* you what to do—to master Flash, you have to *use* it. This book follows that philosophy by providing examples and tutorials in every chapter, and on every topic we cover. It's another well-known fact that although small examples are fun and can help you learn, it's difficult to apply those examples in the real world when you've finished the book. So, at the end of each chapter, you'll be able to apply the things you've just learned to the **case study project**. Each case study is an opportunity for you to put the information you learn in each chapter into building a complete and fully functional website that you can use as your online portfolio:



The website you'll create will have a full navigation menu and animated content, and it will dynamically load in images and text files using ActionScript. As it's created, this real-world example will reinforce the core skills you learn in this book.

We believe that by learning the Flash skills you need in context, you'll build the knowledge and mental adaptability to fit your expanding knowledge and specialization into a structured and reliable framework.

How to use this book

To use this book, all you'll need is a copy of Macromedia Flash 8 and a computer to run it on. This book is written for either Flash Basic or Flash Professional. Most of the book can be completed using only Flash Basic, but a small subset of the features we will discuss are specific to Flash Professional. If you have Flash Basic installed, we recommend that you revisit the Flash Professional specific sections using the 30-day Flash 8 trial available at `www.macromedia.com` (the trial version allows you to trial either Flash Basic or Flash Professional).

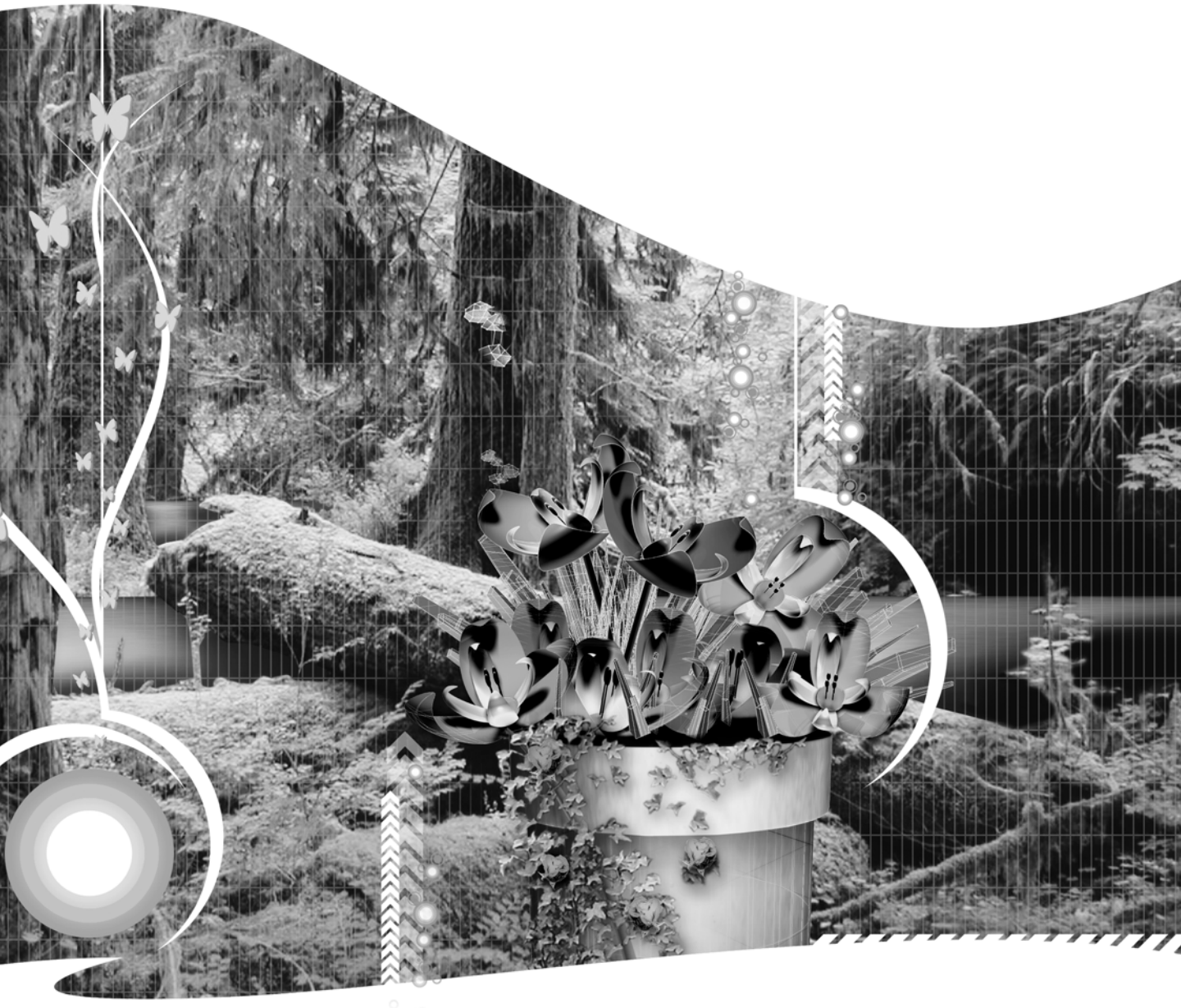
If you want to publish your Flash movies onto the Internet, you'll also need a connection and some web space to publish them to. Your Internet service provider (ISP) will be able to sort this out for you if you have any problems.

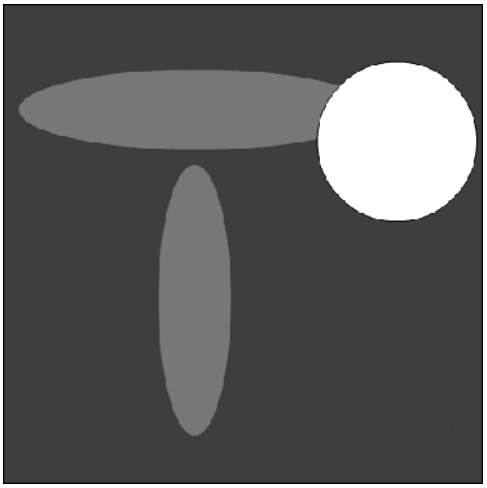
The case study you'll create contains an animated introduction, interactive buttons, and examples of dynamic masking using ActionScript, and it will be fully optimized for publishing to the Web. Its modular nature means that you can easily go back and find the specific functionality that you're looking for and modify it or replace it with something completely different. If, for example, you want to use the buttons in a different website or reuse any of the animated effects, you can easily flip to the relevant chapter for a recap on how to do it, and then just pull the desired part out of the one movie and incorporate it in the other.

You don't have to download anything to use this book, but we've supplied support files containing the sounds and images that we've used to allow you to re-create the examples exactly as they are in the book. The case study files, and all support material, can be found available for download on this book's page at `www.friendsofed.com`. We'll point you to the relevant files in the chapters as necessary.

The case study project files are there so you can pick up the project at any stage in the book and work through it, or you can use them as backups if you've lost your files and don't want to have to re-create them all again. You may just want to check that your results are the same as ours. The files are arranged so that you have a pre-prepared project as the starting point for any chapter. For example, if you want to start from Chapter 5, you'd go into the appropriate folder and use the `for chapter5.fla` file, which contains all the work done on the case study from the beginning through to Chapter 4. Likewise, if you've just finished the Chapter 5 case study and want to check it against ours to make sure it looked right, you would use `for chapter6.fla`.

We also have some optional sound and video files that can be downloaded if you want to try your hand at the compression material we cover in Chapter 11. These (uncompressed) sound and video files are quite large, though, and could take a long time to download on some connections.





Chapter 1

FLASH MOVIE ESSENTIALS

What we'll cover in this chapter:

- Exploring the Flash stage, where you create movies in Flash
- Finding and using tool and object properties in the Properties panel that are guaranteed to make authoring in Flash easier
- Manipulating the size of the stage and changing your movie's overall background in the Document Properties dialog box
- Controlling the playback of your movie in the Timeline
- Using frames to create and arrange the content of your movie
- Making content move through animation
- Using layers to add depth to your movie and keep track of complex content
- Creating separate scenes that contain distinct chunks of your movie

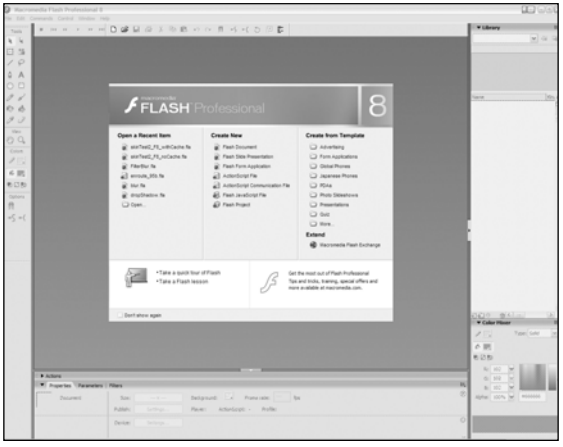
Macromedia Flash is the gateway to state-of-the-art web content. Flash is the standard file format for delivering interactive, visually rich content and animation on the Web. (This is the SWF file format I talked about in the Introduction.) It's also the authoring environment that lets you create and publish the SWF files.

In this chapter, I'm going to introduce you to the authoring environment—the Flash interface—and take you through the essentials of creating visual content in Flash and making it move. In doing this, you'll start building a picture of the main components of a Flash movie and see how they fit together.

Taking time to understand the core elements at the heart of a Flash movie will pay off later. After you have a firm grasp of the foundations, you'll be able to build on them. So let's begin by looking at the first thing that almost everybody wants to do when they open their copy of Flash: create a movie and make interesting things happen on the screen.

The authoring environment

If you don't already have Flash up and running, start it now. When you first open Flash, you see an array of screen elements: icons, menus, toolbars, panels, and status bars.



If you've already explored Flash a little, your screen setup might look a little different from the previous screenshot. The screen may also look slightly different if you are using Flash Basic. For the purposes of this book, it's best to reset to the default setup. Don't worry too much about this for now; I'll show you how to do that in a moment.

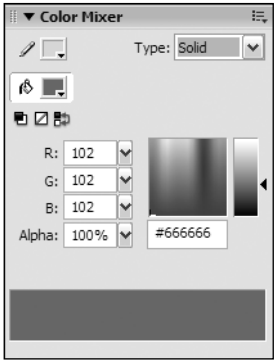
This is the feature-rich authoring environment (Flash's studio, workshop, and test track combined) that lets you create your Flash movies and export them so they can be published on the Web and accessed by the adoring multitudes.

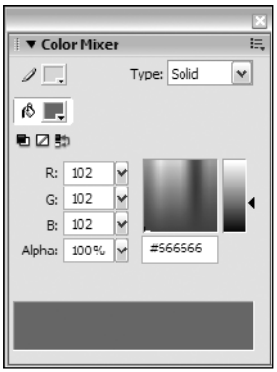
If you've never used a Macromedia product before (and especially if you're coming to Flash from a programming background, where all you're used to is a code editor), you might be intimidated by the unfamiliar interface the first time you open Flash—don't be. Before long, you'll be navigating the interface with ease. And there's an added bonus to learning the Flash interface: Macromedia uses a common interface across all of its software, so once you're familiar with Flash, you'll have no trouble finding your way around other Macromedia programs, such as Dreamweaver.

There's a tremendous amount of detail and power tucked away in the Flash interface, and at first it can seem a little daunting if you're new to the software. To avoid a sense of clutter, and to turn down the volume a little, let's clear some of the elements out of the way. Then you can concentrate on the bare essentials. (Don't worry, you won't be missing a thing—I'll explain all the core features over the course of the book.)

Configuring the authoring environment

1. Select Create New ➤ Flash Document from the central window or File ➤ New ➤ Flash Document.





Flash 8 Professional users will see a number of available document types in the center column of the startup screen. Discussion of most of these document types is beyond the scope of this book (although some of them are covered in the upcoming friends of ED book Foundation Flash 8 ActionScript, expected to be published in 2006). For all the exercises throughout this book, always select the Flash Document *type.*

2. If you've used Flash before picking up this book, reset the default setup by selecting Window ➤ Workspace Layout ➤ Default.

Panels are used to help you modify and manipulate the content of your Flash movie. This content can be graphic images, pieces of animation, text, or code. You can use the panels and menu options to alter their characteristics and the way they behave. All the panels are dockable; you can drag them around the screen and dock them to the other panels. Currently, the panels you can see are docked. An undocked panel is called a floating panel.

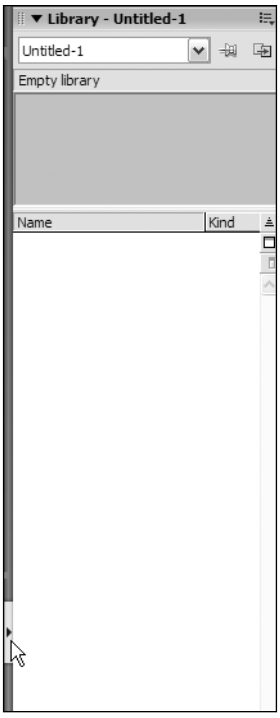
A lone floating panel has a slightly different appearance than a docked one. As shown in the following image, the floating panel (on the right) has an extra bar at the top that the docked panel (on the left) doesn't.

For the benefit of clarity, the panels in screenshots throughout the book are often shown as floating panels. This enables you to clearly focus on that single panel.

Now let's customize the authoring environment. If you decide at a later stage that Macromedia's preset Flash panel layout (Window ➤ Workspace Layout ➤ Default) isn't particularly to your liking, you have the option of saving your own customized layout (Window ➤ Workspace Layout ➤ Save Current).

Each element you create in your movie (such as pictures and text) is treated as a discrete object. Each element has its own attributes, such as color, transparency, and size, and you can use the panels to change these attributes. Additionally, changing panel settings can alter the way an object behaves. You'll look at all of these attributes as you progress through the book. At the moment, though, you don't have any content to work with, so let's move the panels out of the way.

Windows users: To minimize the panel, click the arrow button on the side of the panel or the panel title. To close/open a panel docking area, click the arrow to the side of the docking area as shown in the following image.



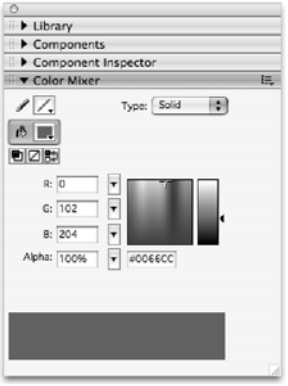
You can also click the arrow belonging to the panels on the bottom of the screen to hide the Properties and Actions panels. This hides the panels from view and provides you with a great deal more screen space to play with.

Note that the panels shown in the following screenshot are docked. A docked panel is one that's fixed in one of the docking areas. There are two docking areas that are most commonly used: the one at the bottom of the screen that shows the Actions and Properties panels, and the one to the far right of the screen that holds the Color and Library panels. Don't worry too much about these panels at the moment (although you should be able to pick them out by looking at their titles).

There are also two other docking areas to the left and top of the screen, which aren't often used. They contain the toolbar panel and Timeline panel, and they're sufficiently important that they should be left as they are, unless you have a really good reason to change them.

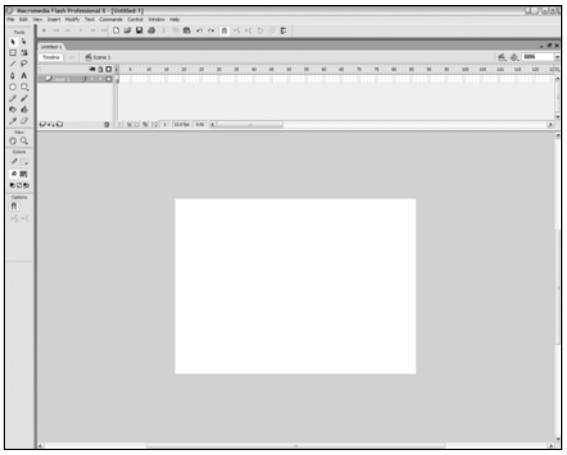
To redisplay the panels, click either of the arrow buttons. Although you've put them out of the way for now, you'll come back to them later.

Mac users: You can close floating panels (i.e., those that are not docked to one of the docking areas noted previously) by clicking the triangle in the upper-left corner of the panel.



After you've hidden the Properties and right-hand panels, you're left with a white area in the middle of the main window. This area is called the stage, and it's what the end user will see when he or she views your finished Flash site or animation. Hopefully by then, the site will have lost some of its current minimalist appearance!

To the left of the screen is a toolbar containing all the Flash drawing tools. Toward the top of the screen is the Timeline. The Timeline is fundamental to the way Flash creates animation, and you'll examine it in detail later in this chapter.

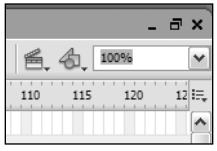


On both platforms, you can close individual panels by right-clicking (Ctrl-clicking on the Mac) the title bar and clicking Close Panel*, or choosing the* Close Panel *option from the drop-down menu (at the upper-right corner of each panel) when the panel is maximized.*

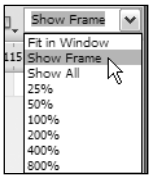
Alternatively, selecting Window ➤ Hide Panels *or pressing F4 will also close all onscreen panels (including the Tools panel). You can bring them back by selecting* Window ➤ Show Panels *or pressing F4 again. Most designers prefer the F4 route because it's the quickest way to clear everything out of sight. F4 is indeed one key to remember.*

Next, you want to make sure that you can see all of the stage area. Being able to fill the screen with the stage area is important because, as you've seen, it's where all your visual content will be created.

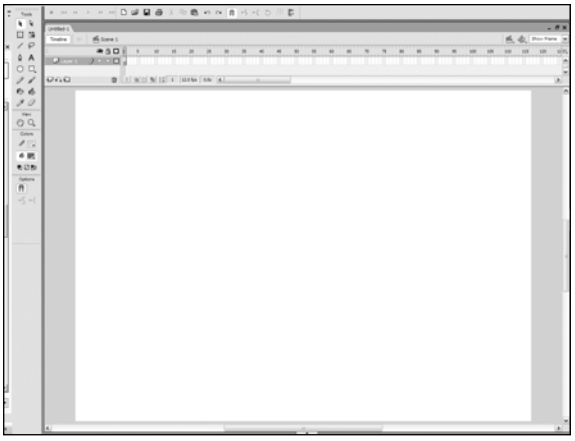
3. Click the Magnification drop-down box near the upper right of the screen above the Timeline.



4. Click the Show Frame option.



The white area in the center of the screen should now be visible in its entirety.



As I've already mentioned briefly, the white area is called the stage, and it's where all the action in your movie takes place. The gray area around the stage is called the work area. Let's talk a little about these different areas.

The stage

The stage can be likened to what a movie director sees in the viewfinder of a camera. What the director sees in the viewfinder is what will appear onscreen when the audience views the finished product. In the motion picture world, the action takes place on the film set (the stage), while actors are waiting offstage, ready to make their entrance.

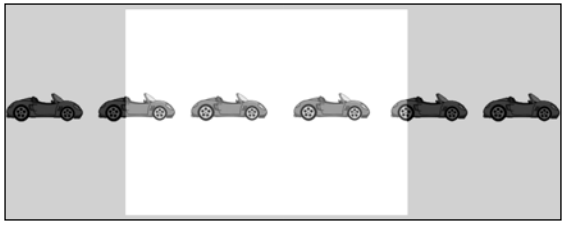
At various times in the movie, different people and objects will be visible on the stage, and consequently in the shot. The stage in Flash works on the same principle: at any given point in your Flash movie, the things that are on the stage are what the viewers will see when the movie is rendered in their browsers. Keep in mind that the movie set can be much larger than the camera's field of view, and the camera can move around and seek out previously hidden corners.

If you want end users to see something in the Flash movie that plays in their browser, that something has to be visible on the stage area when you create the movie. This also means that movie content can move onto the stage from the "wings." For example, an animated actor could enter stage left, walk across the stage, and exit stage right. In the Flash authoring environment, any visual element that moves beyond the boundaries of the stage winds up in the work area.

The work area

The work area surrounds the stage. You can place content in the work area, but usually only content that actually appears on (or moves across) the stage will show up in the finished movie. When you're designing your movie, you need to think about whether the visual elements it contains will spring into existence directly on the stage, or whether they're going to wait in the wings and then move onto the stage at a later time.

One example of this would be a car that starts its journey in the work area to the left of the stage, moves across the stage (and into the viewer's sight), and then accelerates off into the work area on the right.



While the content is on the stage, the viewer sees it in the browser. When it's "in the wings," it's usually invisible to the viewer.

Although content outside the stage usually won't show up in the final Flash page (as seen on the Web), you can force Flash to show content outside the stage area by altering the HTML page that your Flash content is displayed within. For now, though, it's easier to assume that any content outside the stage won't be seen.

So far, you've used the Magnification drop-down box to change your view of the stage *in the authoring environment*. Any changes you make by zooming in and out and making the stage look bigger or smaller in the authoring environment will *not* be applied to the finished movie seen by the end user. These magnification changes are just to help you see things more clearly when you're creating and modifying your movie.

To alter the size and proportions of the stage itself (and therefore your finished movie when it's displayed in the user's browser), you need to change the properties of the stage itself. Flash 8 has a resource that enables you to do that easily—it's called the Properties panel. Let's take a quick look at that now.

The Properties panel

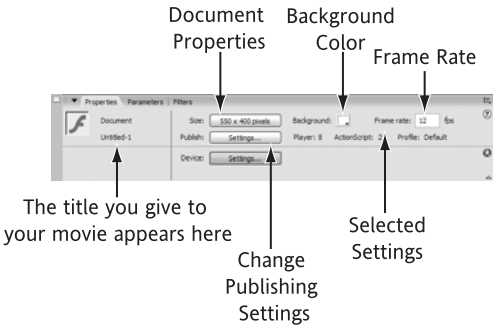
The Properties panel makes working in Flash a whole lot easier, mainly because it's contextual (i.e., the options shown on it will change depending on what you're doing, and in a way such that most of the things you need to do or know are accessible directly from it). With the Properties panel, you can easily manipulate all your movie's contents from one place. You'll be using it frequently throughout this book, and it will come to be your best friend when creating movies in Flash.

By default, the Properties panel is positioned at the bottom center of the screen (you might recall minimizing it when you hid all the panels earlier using *F4*). If you've followed my lead and hidden it, you can retrieve it by clicking the arrow at the bottom center of the screen. You can also press *F4* again to bring all the docking areas and panels back.

If you don't already have it open, you can access the Properties panel by selecting the Window ➤ Properties

menu option or by pressing *CTRL+F3*. Once you have the Properties panel open, click the Properties tab if it isn't already selected.

You don't have any content on your stage yet, so the Properties panel should appear as shown in the following image. If it doesn't appear like this (e.g., if you have a tool selected by mistake or have been poking at something with your mouse), then select the Selection tool (the first tool on the upper left of the toolbar) and click once on the stage.



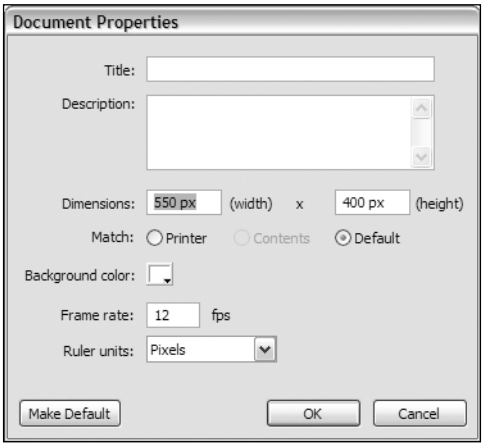
The Properties panel is split into two parts: the upper and lower sections. You'll see in the next chapter that when you use an item from the Tools panel, the upper section has properties specifically related to that tool, whereas the lower section is related to an item you've selected in your work area. The lower section can be opened and closed using the arrow in the lower-right corner.

Let's move on to see how you can use the Properties panel even though you have no content on your stage yet.

The size of the stage

When you're planning your Flash movie, you should consider how much browser space you want the finished movie to take up. You'll need to decide what size you want the movie's window to be, based on factors such as the kind of content you're displaying, what else will appear alongside the movie in the host page, and so on. When you've made that decision, you can alter the size of the stage to match your plan.

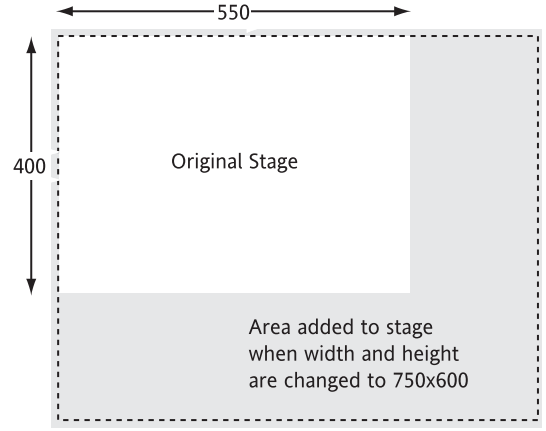
You can view the stage's current dimensions and global characteristics by clicking the Size button in the Properties panel. This opens the Document Properties dialog box, in which you can make global changes to the properties (or characteristics) that affect the whole movie. You can also use it to create a title and comments about the movie.



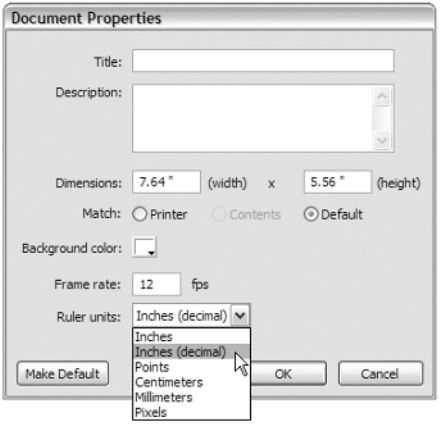
You can also display the Document Properties dialog box by selecting the Modify ➤ Document *menu option, by right-clicking the stage and selecting* Document Properties *from the context-sensitive menu, or by double-clicking the box that displays* 12.0 fps *(located under the Timeline).*

The first thing you can add to the document properties is a title and description. Neither of these is used in the final movie at all—they're just there for you to keep track of what you've created. For now, give the movie a title of My first Flash movie and leave the description blank.

You can see that the default Dimensions of the stage are 550 pixels wide by 400 pixels high. When you change the dimensions of your stage, Flash will always measure them from the upper-left corner. For example, if you change the width of your movie to 750 pixels and the height to 600 pixels, Flash will simply add 200 pixels to the right side of the stage and 200 pixels to the bottom. Flash uses the upper-left corner as the place that the document grows from because this is the convention in print design—much of web design takes its inspiration from print-based graphic design.



If your brain doesn't translate pixel-speak easily, you can always change the units of measurement that Flash uses throughout the entire movie by picking a different measurement option from the Ruler units drop-down box.



Whichever option you choose will be applied throughout the movie until you choose a different option.

When you set the Dimensions *of the stage in the Document Properties dialog box, you're directly affecting the size of the window in which your Flash movie will be displayed on the user's browser. It's good practice to think about this before you start creating your visual content on the stage. You can always change the size of the stage however and whenever you like—but the more advance planning you do, the smoother your movie-creation process is likely to be.*

Note that the Match: Printer and Match: Contents options will automatically change the size of the stage if you select them. Match: Printer will set the stage size to reflect the default paper size for your default printer, and Match: Contents will change the stage so that it's large enough to contain all the content elements you've created (even those that spill over into the work area outside the stage). The Match: Contents option will probably be ghosted out at the moment because you don't have any content on the stage.

There are a number of other global properties that you can change for the whole movie. All the important properties are covered as they come up during the course of building the example movies in this book. At this stage, let's just observe that the Frame rate property influences the playback speed and smoothness of your movie. The default frame rate is 12 fps, which means that your movie will play back at a speed of 12 frames per second (fps). This default frame rate is a trade-off between computer speed and movie smoothness that seemed to work in the early days of Flash. Most designers today tend to set it a little higher, given that computers today can easily handle faster frame rates (and also because Flash itself is better written to optimize animations and can therefore play them faster). A good rule of thumb is 18 fps for new designs for which you want smoothness, although some designers opt for 24 or 30 fps. For now, stick to the default of 12 fps (but feel free to come back later and experiment when you've completed the chapter; the frame rate can be anything from 0.01 to 120 fps).