CHAPTER 3

Planning Flash Projects

✦ ✦ ✦ ✦

In This Chapter

Knowing your workflow

Managing Flash content production

Creating a functional specification document

Using the Project panel in Flash MX Pro 2004

✦ ✦ ✦ ✦

One of the most important steps — if not *the* most important step — to producing great Flash content is *knowing* what steps you'll have to take to move from the concept or idea of the Flash movie to the finished product. This chapter explores the basics of Flash production and how to use the new Project panel in Flash MX Pro 2004 to organize your files. Whether you're a freelance Web consultant (or designer) or a member of a large creative or programming department, knowing how to manage the Flash content production will save you plenty of headaches, time, and money.

Note

The Project panel is available only in Flash MX Professional 2004 (also known as Flash MX Pro 2004). If you are using the trial version of Flash MX 2004 and want to try out the Project panel, make sure to switch to the Professional edition in the Help menu (Help ➪ Switch to Flash MX Professional 2004).

Workflow Basics

No matter what the size or scope, every project in which you choose to participate should follow some type of planned workflow. Whether it's for print, film, video, or Web delivery (or all four!), you should establish a process to guide the production of your presentation.

Before we can explore the way in which Flash fits into a Web production workflow, we need to define a holistic approach to Web production in general. Figure 3-1 shows a typical example of the Web production process within an Internet production company.

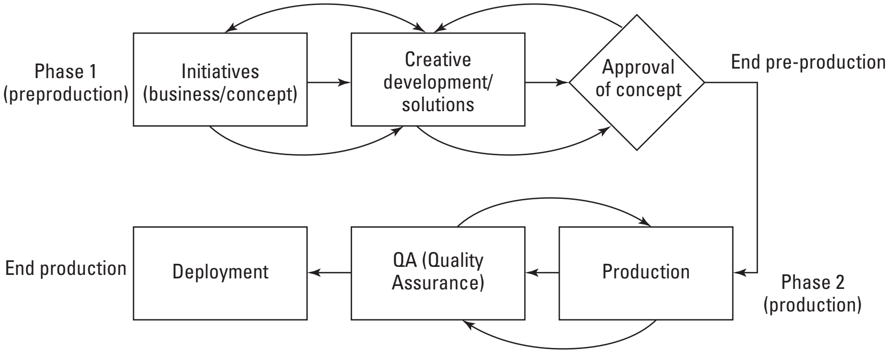


Figure 3-1: Two of the most common phases to generating Web content are preproduction and the actual production phase.

Phase I: Establishing the concept and goals

As a Web developer or member of a creative team, you will be approached by companies (or representatives for other departments) to help solve problems with projects. A problem may or may not be well defined by the parties coming to you. The goal of Phase I is to thoroughly define the problem, offer solutions for the problem, and approve one (or more) solutions for final production.

Defining the problem

Before you can help someone solve a problem, you need to determine what the problem is and whether there is more than one problem. When we say *problem,* we don't mean something that's necessarily troublesome or irritating. Think of it more as a math problem, where you know what you want — you're just not sure how to get there. When you're attempting to define a client's problem, start by asking them the following questions:

- ✦ What's the message they want to deliver? Is it a product that they want to feature on an existing Web site?
- ✦ Who's their current audience?
- ✦ Who's their ideal audience? (Don't let them say, "Everyone!")
- ✦ What branding materials (logos, colors, and identity) do they already have in place?
- ✦ Who are their competitors? What do they know about their competitors?

The last question points to a bigger picture, one in which the client may already have several emotive keywords that define their brand. Try to define the emotional heart and feeling of their message — get them to be descriptive. Don't leave the meeting with the words *edgy* or *sexy* as the only descriptive terms for the message.

Tip

Never go into a meeting or a planning session without a white board or a big pad of paper. Documenting everyone's ideas and letting the group see the discussion in a visual format is always a good idea. If all participants are willing, it's often useful to record the meeting with a digital voice recorder or video camera, so that it can be reviewed outside of the meeting.

Information Architects

You may have already been bombarded with the idea of *information architecture*. Information architecture is the method by which sought data is represented and structured. Good information architecture is usually equivalent to intuitive user interface design—visitors to a well-organized Web site won't spend much time finding what they came for.

We mention information architecture because the steps in Phase I are similar to the steps that traditional architects take to build a comprehensive design and production strategy *before* they start building any structure. Although this may seem obvious enough, the sad fact remains that most Internet sites (or projects) are planned as they're constructed. Indeed, we're told that production must move at Internet speed—directives can be given without thorough research into other solutions to the problem.

You can also start to ask the following technical questions at this point:

- What type of browser support do you want to have?
- Do you have an idea of a Web technology (Shockwave, Flash, DHTML, SVG) that you want to use?
- Does the message need to be delivered in a Web browser? Can it be in a downloadable application such as a standalone player? A CD-ROM? A DVD?
- What type of computer processing speed should be supported? What other types of hardware concerns might exist (for example, hi-fi audio)?

Of course, many clients and company reps look to *you* for the technical answers. If this is the case, the most important questions are:

- Who's your audience?
- Who do you *want* to be your audience?

Your audience determines, in many ways, what type of technology to choose for the presentation. If they say that Ma and Pa from a country farm should be able to view the Web site with no hassle, then you may need to consider a non-Flash presentation (such as HTML 3.0 or earlier), unless it's packaged as a standalone player that's installed with a CD-ROM (provided by the client to Ma and Pa). However, if they say that their ideal audience is someone who has a 56K modem and likes to watch mature cartoons, then you're getting closer to a Flash-based presentation. If they have any demographic information for their user base, ask for it up front. Putting on a show for a crowd is difficult if you don't know who's *in* the crowd.

Cross-Reference

Stand-alone players are discussed in Chapter 23, "Using the Flash Player and Projector."

Determining the project's goals

The client or company rep comes to you for a reason—they want to walk away with a completed and *successful* project. As you initially discuss the message and audience for the presentation, you also need to get a clear picture of what the client expects to get from you.

- Will you be producing just one piece of a larger production?
- Do they need you to host the Web site? Or do they already have a Web server and a staff to support it?
- Do they need you to maintain the Web site after handoff?
- Do they expect you to market the presentation? If not, what resources are in place to advertise the message?
- When does the client expect you to deliver proposals, concepts, and the finished piece? These important dates are often referred to as *milestones*. The payment schedule for a project is often linked to production milestones.
- Will they expect to receive copies of all the files you produce, including your source FLA files?
- What are the costs associated with developing a proposal? Will you do work on speculation of a potential project? Or will you be paid for your time to develop a concept pitch? (You should determine this *before* you walk into your initial meeting with the client.) Of course, if you're working with a production team in a company, you're already being paid a salary to provide a role within the company.

At this point, you'll want to plan the next meeting with your client or company rep. Give them a realistic timeframe for coming back to them with your ideas. This amount of time will vary from project to project and will depend on your level of expertise with the materials involved with the presentation.

Creative exploration: Producing a solution

After you leave the meeting, you'll go back to your design studio and start cranking out materials, right? Yes and no. Give yourself plenty of time to work with the client's materials (what you gathered from the initial meeting). If your client sells shoes, read up on the shoe business. See what the client's competitor is doing to promote their message — visit their Web site, go to a store and compare the products, and read any consumer reports that you can find about your client's products or services. You should have a clear understanding of your client's market and a clear picture of how your client distinguishes their company or their product from their competitors'.

After you (and other members of your creative team) have completed a round of research, sit down and discuss the findings. Start defining the project in terms of mood, response, and time. Is this a serious message? Do you want the viewer to laugh? How quickly should this presentation happen? Sketch out any ideas you and any other member of the team may have. Create a chart that lists the emotional keywords for your presentation.

At a certain point, you need to start developing some visual material that articulates the message to the audience. Of course, your initial audience will be the client. You are preparing materials for them, not the consumer audience. We assume that you are creating a Flash-based Web site for your client. For any interactive presentation, you need to prepare the following:

- An organizational flowchart for the site
- A process flowchart for the experience
- A functional specification for the interface
- A prototype or a series of comps

An *organizational flowchart* is a simple document that describes the scope of a site or presentation. Other names for this type of chart are *site chart, navigation flowchart,* and *layout flowchart.* It includes the major sections of the presentation. For example, if you're creating a Flash movie for a portfolio site, you might have a main menu and four content areas: about, portfolio, resume, and contact. In an organizational flowchart, this would look like Figure 3-2.

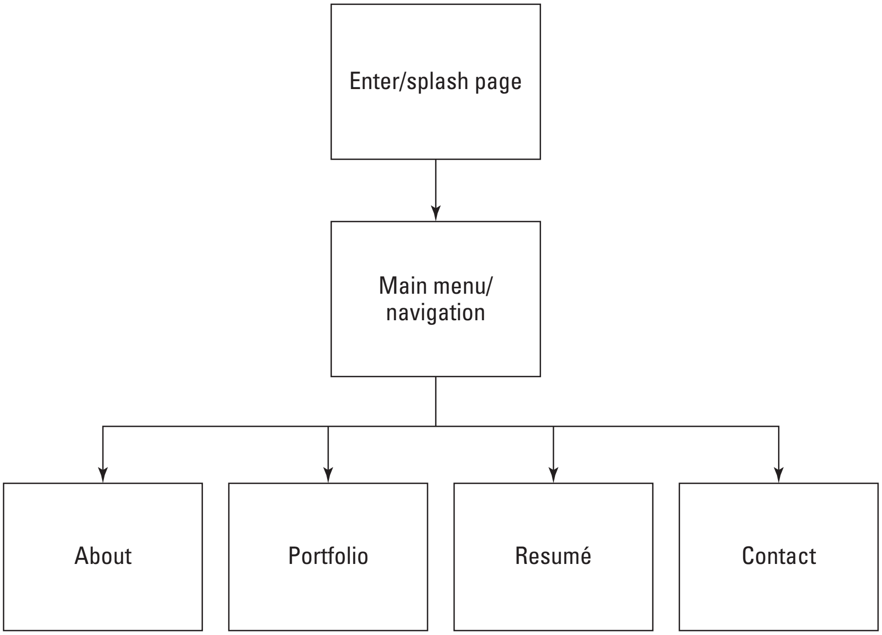


Figure 3-2: A sample organizational chart for a portfolio site

A *process flowchart* constructs the interactive experience of the presentation and shows the decision-making process involved for each area of the site. There are a few types of process charts. A basic process flowchart displays the decision-making of the end-user (for example, what type of options does a user have on any given page of the site?). Another type of flowchart shows the programming logic involved for the end-user process chart. For example, will certain conditions need to exist before a user can enter a certain area of the site? Does he have to pass a test, finish a section of a game, or enter a username and password? Refer to Figure 3-3 for a preliminary flowchart for a section of our portfolio Web site. We discuss the actual symbols of the flowchart later in this chapter.

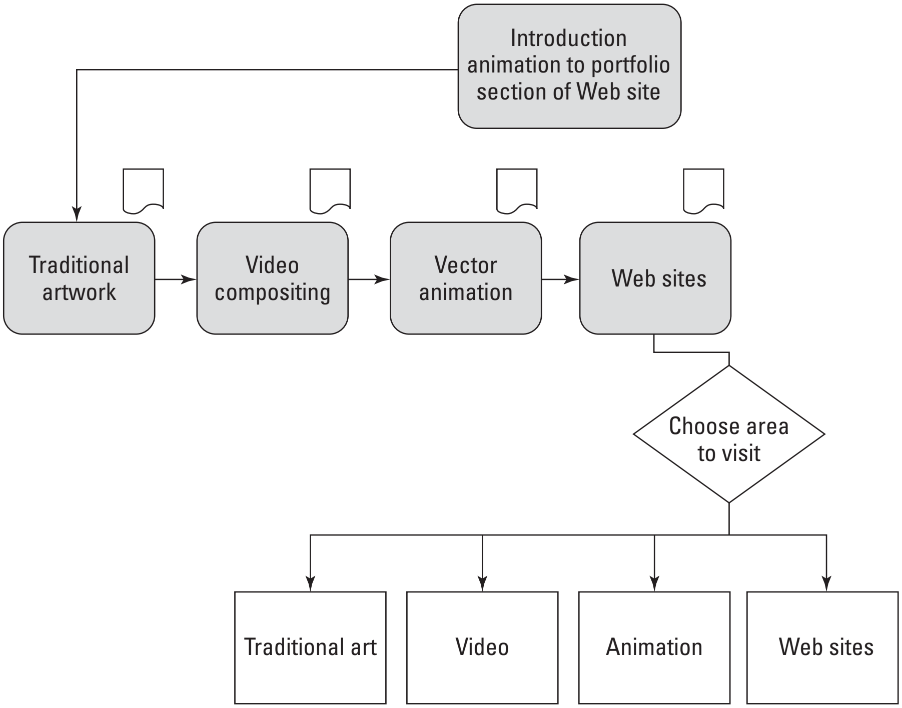
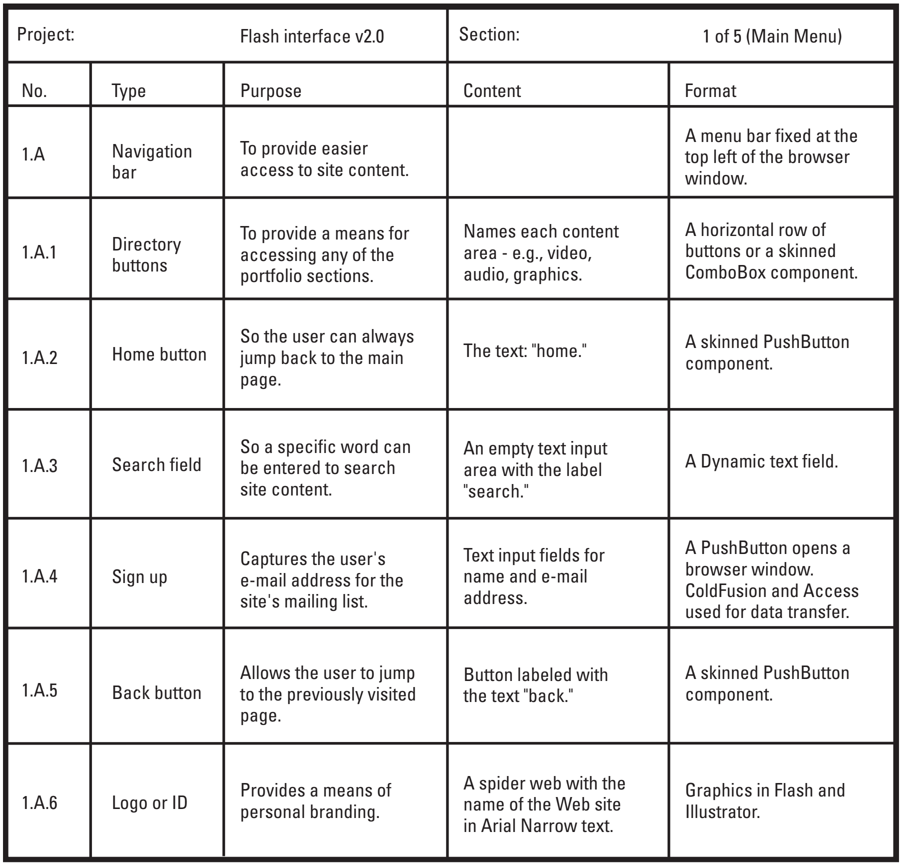


Figure 3-3: The user watches an intro animation and is led through several short subsequent animations detailing each area of the portfolio. The user can then go to an area of his choice after this animation is complete.

A *functional specification* (see Figure 3-4) is a document that breaks down the elements for each step in the organizational and/or process flowchart. This is by far the most important piece of documentation that you can create for yourself and your team. Each page of a functional specification (functional spec, for short) lists all the assets used on a page (or Flash scene, keyframe, Movie Clip) and indicates the following information for each asset:

- ✦ **Item ID:** This is part of the naming convention for your files and assets. It should be part of the filename, or Flash symbol and instance name. It should also be used in organizational and process flowcharts.
- ✦ **Type:** This part of the spec defines the name you're assigning to the asset, in more natural language, such as Home Button.
- ✦ **Purpose:** You should be able to clearly explain why this element is part of the presentation. If you can't, then you should consider omitting it from the project.
- ✦ **Format:** This column indicates what technology (or what component of the technology) will be utilized to accomplish the needs of the asset. In an all-Flash presentation, list the symbol type or timeline component (frames, scene, nested Movie Clips) necessary to accomplish the goals of the asset.



Project:		Flash interface v2.0	Section:	1 of 5 (Main Menu)
No.	Type	Purpose	Content	Format
1.A	Navigation bar	To provide easier access to site content.		A menu bar fixed at the top left of the browser window.
1.A.1	Directory buttons	To provide a means for accessing any of the portfolio sections.	Names each content area - e.g., video, audio, graphics.	A horizontal row of buttons or a skinned ComboBox component.
1.A.2	Home button	So the user can always jump back to the main page.	The text: "home."	A skinned PushButton component.
1.A.3	Search field	So a specific word can be entered to search site content.	An empty text input area with the label "search."	A Dynamic text field.
1.A.4	Sign up	Captures the user's e-mail address for the site's mailing list.	Text input fields for name and e-mail address.	A PushButton opens a browser window. ColdFusion and Access used for data transfer.
1.A.5	Back button	Allows the user to jump to the previously visited page.	Button labeled with the text "back."	A skinned PushButton component.
1.A.6	Logo or ID	Provides a means of personal branding.	A spider web with the name of the Web site in Arial Narrow text.	Graphics in Flash and Illustrator.

Figure 3-4: This functional spec displays the six components of a Flash-based navigation bar, which will appear on the main menu of our portfolio content site.

Note

Functional specs come in all shapes and sizes. Each company usually has their own template or approach to constructing a functional spec. The client should always approve the functional spec, so that you and your client have an agreement about the scope of the project.

Finally, after you have a plan for your project, you'll want to start creating some graphics to provide an atmosphere for the client presentation. Gather placement graphics (company logos, typefaces, photographs) or appropriate "temporary" resources for purposes of illustration. Construct one composition, or *comp*, that represents each major section or theme of the site. In our portfolio content site example, you might create a comp for the main page and a comp for one of the portfolio work sections, such as "Animation." Don't create a comp for each page of the portfolio section. You simply want to establish the feel for the content you will create for the client. We recommend that you use the tool(s) with which you feel most comfortable creating content. If you're better at using FreeHand or Photoshop to create layouts, then use that application. If you're comfortable with Flash for assembling content, then use it.

Caution

Do not use copyrighted material for final production use, unless you have secured the appropriate rights to use the material. However, while you're exploring creative concepts, use whatever materials you feel best illustrate your ideas. When you get approval for your concept, improve upon the materials that inspired you.

Then you'll want to determine the time and human resources required for the entire project or concept. What role will you play in the production? Will you need to hire outside contractors to work on the presentation (for example, character animators, programmers, and so on)? Make sure you provide ample time to produce and thoroughly test the presentation. When you've determined the time and resources necessary, you'll determine the costs involved. If this is an internal project for your company, then you won't be concerned about cost so much as the time involved — your company reps will want to know what it will cost the company to produce the piece. For large client projects, your client will probably expect a project rate — not an hourly or weekly rate. Outline a time schedule with milestone dates, at which point you'll present the client with updates on the progress of the project.

Exploring the details of the workflow process any further is beyond the scope of this book. However, there are many excellent resources for project planning. One of the best books available for learning the process of planning interactive presentations is Nicholas Iuppa's *Designing Interactive Digital Media.* We strongly recommend that you consult the *Graphic Artists Guild Handbook of Pricing and Ethical Guidelines* and the *AIGA Professional Practices in Graphic Design,* edited by Tad Crawford, for information on professional rates for design services.

Approving a final concept and budget

After you have prepared your design documents for the client, it's time to have another meeting with the client or company rep. Display your visual materials (color laser prints, inkjet mockups, and so on), and walk through the charts you've produced. In some situations, you may want to prepare more than one design concept. Always reinforce how the presentation addresses the client's message and audience.

Cross-Reference

See Todd Purgason's tutorial on Flash and FreeHand, in Chapter 36, "Working with Vector Graphics." He offers some excellent suggestions for creating high-impact presentation boards in FreeHand and tips on how to reuse graphics in print and Web projects.

When all is said and done, discuss with the client the options that you presented. Gather feedback. Hopefully, the client prefers one concept (and its budget) and gives you the approval to proceed. It's important that you leave this meeting knowing one of two things:

- The client has signed off on the entire project or presentation.
- The client wants to see more exploration before committing to a final piece.

In either case, you shouldn't walk away not knowing how you'll proceed. If the client wants more time or more material before a commitment, negotiate the terms of your fees that are associated with further conceptual development.

Designing for Usability, by Scott Brown

As mentioned earlier in this chapter, the first step in developing a Flash site, or any other type of site, is to define the information architecture. In this tutorial, you find out how to define the goals and mission of the site.

Defining the goals and mission of the site

Defining the mission and goals is laying the foundation upon which to build your project. To create a solid project foundation, we must begin by questioning everything, especially the company's business model. Start with these questions:

- What is the mission or purpose of the organization?
- Why does this organization want a Web site?
- Will the Web site support the mission of the organization?
- What are the short and long-term goals of the Web site?
- Who are the intended audiences?
- Why will people come to the site?
- Are we trying to sell a product?
- What is the product or products?
- Do we have a unique service?
- What makes the service different?
- Why will people come to the site for the first time?
- Will they ever come back?
- Why would they come back?

The list of questions can go on forever. After you have gathered a list of questions, you need to get the answers. Ask around the organization, ask your friends, ask strangers, ask anyone. After the answers have been collected, you need to filter through them to create a list of goals that are based on the responses. From this list of goals, you must define further the answer to the question, "Who is the audience?"

Defining the audience

The audience can be defined as the potential users of the site and by their intentions or tasks that they may have when they come to your site. Are they kids or adults? Are they Generation X, Y, or Z? Are they into rave music or country music?

So, who is your audience? It's not an easy question because there are so many possibilities. Start with a list of all the possible audiences that the organization would like to reach, and then rearrange the list in a ranking order of most important audience to least important audience. From the audience-ranking list, create a list of possible goals and needs of each audience.

Creating character scenarios

With the list of possible goals take the process one step further by creating scenarios of the users. Think of it as writing a screenplay for your Web site. Create multiple characters that represent the majority of visitors with hobbies, likes, dislikes, and, most importantly, a task to complete on the site. The object of the scenario game is to get into the characters' heads to learn why and how they would use your site. From their viewpoint you will have an easier time creating a list of needs and wants for the character, a wish list if you will.

After the scenarios are written, the next step in the process is to gather the team together and analyze the Web sites of the competition.

Analyzing the competition

Studying the competition gives you the chance to generate a list of what kind of features they are offering and to determine whether your feature list, the one that you created from the scenarios, is missing anything. If your wish list is lacking anything in comparison to your competition, now is a good time to expand the user's functionality requirements, and to return to the scenarios to determine whether the competition's functionality matches your character's needs. If it does, you should try to elaborate on their functions and create new functions of your own — the classic case of outdoing your competition.

Reaching a consensus on what good design is

At this time in the process, have the team come together to develop a definition of what is "good site design." This step is most beneficial for any contract designer trying to gain an understanding of the client's design viewpoint. To create this "good design" definition, the team should observe a good number of sites and document everybody's likes and dislikes for each site. From the documentation of this exercise, everyone on the team will have a better understanding of what to strive for and what to avoid.

Structuring the content

Now you should have several documents to refer to — the project mission statement, the user functionality needs (wish list), and the organization's definition of good design. With these three documents in hand, the next step is to blend them into one master menu of content inventory. Think of each item on this list as a building block. You now have all the blocks needed to construct the site. The only problem is that these blocks are in a big pile with no organization (structure). Naturally, the next step is to begin creating layouts of the site, providing structure. But before you can begin the page layout process, you need to educate yourself on some Web site usability issues.

Identifying factors of usability

Usability is a much-debated concept, but generally it means creating a site/project/interface that is functional and that your audience understands. A usable site aims to be a natural extension of a user's expectations and needs. A user-friendly site tries to mirror its structure to that of the user's experience and goals. Just to make the task at hand a little more complex, keep in mind that user expectations learned in other areas of life affect how the user will think your site works. So, how can you design a site to meet your user's expectations? Well, if you did your homework on your audience and wrote the character scenarios, you should have a pretty good idea of the target audience's expectations. By knowing the general background of a user, you could include metaphors into the structure of the site. Using metaphors is a great way to help users draw upon knowledge they already have, thereby making the site easier to use. By matching the site structure to the user's experience, the amount of time it takes for the user to learn how to operate or navigate the site is minimized. The shorter the learning curve for the site, the better. If you come to a site when you have a specific goal in mind, and it takes you ten minutes to figure out how to achieve your goal, would you call that a positive experience? Most likely not!

The goal of the designer is to create an attractive site without distracting the users from their goals. Forcing the users to spend a noticeable amount of time trying to learn how to achieve their goals is very taxing on their patience, and is a good way to create a negative experience. If you're trying to sell something, chances are you want customers to be happy not annoyed. One way to make your customers' experience more enjoyable is to make their experience as easy as possible. So, how do you create a positive experience? Let's start with the most basic of user needs: the ability to navigate.

Users need to know at all times where they are in the site, where they have been, and where they can go. When developing a navigation system, be sure to keep the navigation visually consistent. Inconsistency in the navigation can cause the user to be confused and frustrated. A great concept for a navigational aid is the use of a breadcrumb trail. The breadcrumb system is a visual way to show the user the path they took to get to their current position in the site. This navigational convention is used on many resource sites and even in the Flash authoring environment itself—as you click in to edit grouped shapes or symbols, the steps you've taken are shown as text labels on the bar above the Stage. Beyond displaying the path of the user, this system gives the user the ability to backtrack to any page displayed in the path. However, remember that navigation is not the goal of the user, only an aid. The user is there to find or buy something; the user is there for the content. So, make the content the first read on all your pages. Navigational elements are there to support the content, not eclipse it.

Of course, navigation isn't the only factor to consider when designing for usability. Other variables, such as the length of text on a page, can affect the usability of a site tremendously. It's a fact that reading text on a monitor is far more taxing on the eyes than reading text on paper. Therefore, people are less inclined to read large amounts of text on the Web. As designers, we must accommodate these changes in reading patterns. Keep these simple guidelines in mind when writing text for the Web. Try to make the text scannable, because readers skim Web content. Bold the important ideas or put key information in bulleted lists. But most of all, keep the text short.

In addition to the treatment of text, there are several other tips to help improve the usability of a site. The concept of redundant links is an excellent method to support users with different backgrounds and goals. With redundant links, a user has more than one way to get to the desired content. The user may have the option to click a text link, a graphic link, or even a text link that is worded differently. Each redundant link should be designed to accommodate a wide range of users. So, where on the page should all these usability elements go?

I can't tell you where you should place your navigation system or your redundant links. However, I can provide you with some information on eye-tracking studies that will help you make an educated decision. Yes, it is true that usability researchers are able to actually monitor and record what you're looking at when viewing a Web site. Researchers have found that when a Web page loads, our eyes are looking at the center of the page, then move over to the left, and then sometimes to the right. Of course, these findings are dependent on the user's cultural background. Nevertheless, the scary finding is that the users rarely look to the right! This is most likely because most sites use the right side of the page as a place to add sidebar elements, items of lesser importance. This is also a good example of how a user's experience can affect his future experiences. So, how does Flash fit into Web site usability considerations?

Flash is a great design tool to create amazing interfaces. Flash gives the designer the freedom to create almost anything he desires. But the flexibility given to the designer is also Flash's greatest weakness from a usability perspective. Flash is a great tool for creating animation, however, inexperienced Web designers can easily go overboard. Just because you can animate an object doesn't mean that you should. The eye is very sensitive to the smallest amount of animation or movement in its peripheral view, pulling the eyes' attention away from the site's main content. On the plus side, animation used as a transitional element is very beneficial for the user. Animated transitions enable the user to follow the navigation process, gaining a better understanding of how the site might work.

Along with the problems of animation abuse, Flash enables the designers to create their own graphical user interface (GUI) elements. This is great for the designers, but the users are often left out in the cold with all this newfound freedom. This design freedom is forcing the user to learn, almost from scratch, how to operate a scroll bar or a navigation bar. If you recall, earlier we mentioned the importance of a short learning curve for the users. These extreme creative versions of standardized GUI elements might rank high on the "cool" scale, but they really throw a monkey wrench into the user's goal and expectations. GUI standards are developed to help create a consistent experience across all platforms in an effort to eliminate any unpleasant surprises. Again, these usability problems can be avoided in Flash by educating the designers about the issues at hand and finding solutions based on the set standards.

Other usability issues with Flash arise from the actual plug-in nature of Flash. Unfortunately, because Flash requires a plug-in to work in Web browsers, Flash movies are unable to take advantage of some of the browser's built-in capabilities such as the Back button and the capability to display history for the links by changing the color of the links that have been clicked. The problem with the browser's Back button is that when the button is pressed, the browser takes the user back to the previous HTML page, not to the previous state in the Flash movie. It's not a nice surprise for unsuspecting users. One solution to this problem is to pop up the Flash movie in a new browser window (via JavaScript) with all the browser's navigation elements removed (in other words, no toolbar, no location bar, no menus, and so on). No Back button on the browser, no problem right?

Cross-Reference

Fortunately, ActionScript can often smooth out the discrepancies between the default browser behaviors and Flash movie functionality. For an introduction to some ActionScript solutions, refer to Part VI, "Distributing Flash Movies."

Tip

Named anchor keyframes, available in Flash Player 6 or higher movies, is a feature that makes it easier for users to navigate a Flash movie using keystrokes. For an example of named anchors in action, refer to Chapter 20, "Making Your First Flash MX 2004 Project."

Building mockups of the site

You are now ready to begin mocking up the site structure using index cards, sticky notes, and other common office supplies. Creating these paper mockups saves the development team a large amount of time. The beauty of the paper mockups is that you can quickly create a navigational system and find the major flaws without spending long hours developing a beautiful rendering of a structure that may be flawed. There is nothing worse than spending months developing a product with a faulty structure only to discover the mistake just before launch!

Testing the site on real users

Testing the site is the most important step in creating a usable site. The key to testing the site is *not* to test it on people of the organization, but to test it on people in the target audience. Test the site on the real users. It's usually easier to test the site by using people who are familiar with the project. The problem with that practice is that the people are familiar with the project. You want to test fresh eyes and minds in order to get optimum feedback. For testing purposes, create a list of several tasks to complete on the site. The tasks should be pulled from the list of possible users' goals defined in the early steps of the project. As the test subjects navigate through your project, pay close attention to how long it takes them. How many times did they have to click to find what they were looking for? How many had to resort to using a search feature (or wished that they could)? What elements seemed to cause confusion or delay? What elements attracted or held the users' attention? After each test subject has completed a task, or tried, give her a post-task questionnaire with questions such as:

"How would you rate the quality of the content on this site?"

Unacceptable –3 –2 –1 0 1 2 3 Excellent

Also, leave some room for the test subject to elaborate on the questions. After the testing is finished, review your findings and determine what needs to be fixed. After the problems are fixed, test the site again, but on new users. Repeat the process until you have a product that meets the defined goals of the organization and the users. Keep asking yourself this question: Is the interface helping the users accomplish their goals? When all else fails, you can always depend on the greatest guideline of the century: keep it simple. Oh, how true.

Tip

If this process is new to you, don't waste time "reinventing the wheel" when there are plenty of resources on this topic that can get you started. Jakob Nielsen has achieved near-celebrity status as a result of his strong opinions on this topic and he has written several books that some consider definitive. Steve Krug and Roger Black have written another popular book on this topic titled *Don't Make Me Think: A Common Sense Approach to Web Usability* (New Riders, 2000).

Phase II: Producing, testing, and staging the presentation

When your client or company executives have signed off on a presentation concept, it's time to rock and roll! You're ready to gather your materials, assemble the crew, and meet an insane production schedule. This section provides a brief overview of the steps you need to take to produce material that is ready to go live on your Web site.

Assembling assets

The first step is to gather (or start production of) the individual assets required for the Flash presentation. Depending on the resources you included in your functional spec and budget, you may need to hire a photographer, illustrator, animator, music composer (or all four!) to start work on the production. Or, if you perform any of these roles, then you'll start creating rough drafts for the elements within the production. At this stage, you'll also gather high-quality images from the client for their logos, proprietary material, and so on.

Making the Flash architecture

Of course, we're assuming that you're creating a Flash-based production. All the resources that you've gathered (or are working to create) in Step 1 will be assembled into the Flash movie(s) for the production. For large presentations or sites, you'll likely make one master Flash movie that provides a skeleton architecture for the presentation, and use `loadMovie()` to bring in material for the appropriate sections of the site.

Cross-Reference

See our coverage on the new Project panel later in this chapter. You can organize several Flash movies with this panel.

Before you begin Flash movie production, you should determine two important factors: frame size and frame rate. You don't want to change either of these settings midway through your project. Any reductions in frame size will crop elements that weren't located near the top-left portion of the Stage—you'll need to recompose most of the elements on the Stage if you used the entire Stage. Any changes in your frame rate will change the timing of any linear animation and/or sound synchronization that you've already produced.

Staging a local test environment

As soon as you start to author the Flash movies, you'll create a local version of the presentation (or entire site) on your computer, or a networked drive that everyone on your team can access. The file and folder structure (including the naming conventions) will be consistent with the structure of the files and folders on the Web server. As you build each component of the site, you should begin to test the presentation with the target browsers (and Flash Player plug-in versions) for your audience.

HTML page production

Even if you're creating an all-Flash Web site, you need a few basic HTML documents, including:

- **A plug-in detection page** that directs visitors without the Flash Player plug-in to the Macromedia site to download the plug-in.
- **HTML page(s) to display any non-Flash material** in the site within the browser.

New Feature

Flash MX 2004 can create detection pages for you. See Chapter 21, "Publishing Flash Movies," for more information.

You will want to construct basic HTML documents to hold the main Flash movie as you develop the Flash architecture of the site.

Staging a server test environment

Before you can make your Flash content public, you need to set up a Web server that is publicly accessible (preferably with login and password protection) so that you can test the site functionality over a non-LAN connection. This also enables your client to preview the site remotely. After quality assurance (QA) testing has finished (the next step that follows), you'll move the files from the staging server to the live Web server.

We've noticed problems with larger SWF files that weren't detected until we tested them from a staging server. Why? When you test your files locally, they're loaded instantly into the browser. When you test your files from a server (even over a fast DSL or cable modem connection), you have to wait for the SWF files to load over slower network conditions. Especially with preloaders or loading sequences, timing glitches may be revealed during tests on the staging server that were not apparent when testing locally.

Tip

You should use the Simulate Download feature of the Bandwidth Profiler in the Test Movie environment of Flash MX 2004 to estimate how your Flash movies will load over a real Internet connection. See Chapter 21, "Publishing Flash Movies," for more discussion of this feature.

Quality assurance testing

In larger corporate environments, you'll find a team of individuals whose sole responsibility is to thoroughly test the quality of a nearly finished production (or product). If you're responsible for QA, then you should have an intimate knowledge of the process chart for the site. That way, you know how the site should function. If a feature or function fails in the production, QA reports it to the creative and/or programming teams. QA teams test the production with the same hardware and conditions as the target audience, accounting for variations in:

- Computer type (Windows versus Mac)
- Computer speed (top-of-the-line processing speed versus minimal supported speeds, as determined by the target audience)
- Internet connection speeds (as determined by the target audience)
- Flash Player plug-in versions (and any other plug-ins required by the production)
- Browser application and version (as determined by the target audience)

Tip

If you're a freelance designer or operate a small company, keep in mind that there is no such thing as a useless computer—recycle your older computers as test platforms for target audiences.

Web Resource

It's worthwhile to use an online reporting tool to post bugs during QA. Many companies use the open source (freeware) tool called Mantis, a PHP/MySQL solution. You can find more information about Mantis at `mantisbt.sourceforge.net`.

After QA has finished rugged testing of the production, pending approval by the client (or company executives), the material is ready to go live on the site.

Maintenance and updates

After you've celebrated the finished production, your job isn't over yet. If you were contracted to build the site or presentation for a third party, you may be expected to maintain and address usability issues provided by follow-ups with the client and any support staff they might have. Be sure to account for periodic maintenance and updates for the project in your initial budget proposal. If you don't want to be responsible for updates, make sure you advise your clients ahead of time to avoid any potential conflicts after the production has finished.

You should have a thorough staging and testing environment for any updates you make to an all-Flash site, especially if you're changing major assets or master architecture files. Repeat the same process of staging and testing with the QA team that you employed during original production.

Web Resource

You can find an online archived PDF version of Eric Jordan's tutorial, "Interface Design," on the book's Web site at `www.flashsupport.com/archive`. This tutorial was featured in the last edition of the Bible. Also, look at `www.flashsupport.com/articles` for a future update to the "Using Microsoft Visio to Create Flowcharts" coverage from the last edition as well.

Using the Project Panel in Flash MX Pro 2004

In this final section of the chapter, you learn how to use the Project panel in Flash MX Pro 2004 with some sample files provided to you on this book's CD-ROM. You'll jump right into the Project panel, so you may want to review some of the content in the Help panel of Flash MX 2004 before proceeding. The following pages in the Help panel (Help ➪ Help) contain useful information about the Project panel:

- Using Flash ➪ Working with Projects overview
- Using Flash ➪ Creating and managing projects

Tip

You can quickly find these pages in the Help panel by searching with the keyword **project**.

Before you start using the Project panel, let's describe a scenario in which you would *want* to use the feature. The Project panel lets you organize and group all of the files related to a Flash production. You can include any file type you want in the Project panel. All of the asset names and locations are stored in a Flash Project file, which uses a .flp file extension. This file is essentially an XML file that describes the files you want to manage.

Once you have a Flash Project file created, you can quickly open any document directly in Flash or another application. You can publish one or more Flash documents in the project. But more importantly, you can use the Project panel to directly upload content to your FTP server or a local network server. The Project panel can check in and check out files, so that other members on your team know that you're working on them.

The Project file is linked to a site definition in the Project panel. The site definition is exactly the same site you may have created in Dreamweaver MX 2004. If you have made a site in Dreamweaver, it is automatically available to Flash MX Pro 2004 as well.

One important factor to kind in mind when you use the Project panel is that you only ever open a local copy of the project's files on your computer. In this way, everyone working on the project has his/her own copy of the files. In this way, one member can be editing, implementing, and testing changes while other members are doing the same with their copies. When a member is done editing a file, she can check the file back into the server.

Caution

Unless you're implementing a version control system with your project files, you should **not** edit the same file that another person is using. Currently, Flash MX Pro 2004 only ships with support for Microsoft SourceSafe, a version control product. You can develop your own plug-in, though, for your particular version control product. Version control systems have the ability to merge changes to the same document. For example, if two people edit the same ActionScript document (.as file), the version control system merges the changes into one file and even flags potential conflicts during the process. The Project panel can not perform this type of merge without the assistance of a separate software product such as SourceSafe. Also, it's important to note that version control software cannot merge changes in two Flash documents (FLA files) because such files are binary, not ASCII (or Unicode). Usually, only text documents can be merged by version control systems.

In the following exercise, let's quickly review the procedure you follow:

1. Establish a site definition in Flash MX Pro 2004. This definition describes where you'll store your local copy of the project files, and where to upload the master copies of the project.
2. Add the files to the Flash Project file in the Project panel of Flash MX Pro 2004.
3. Open, edit, and test one of the sample files.
4. Create a new blank document to add to the project.
5. Publish an entire project.

In order to easily reference each of these procedures in the exercise, look for the procedure number from the previous list in the respective heading of the following sections.

1. Establishing a project and a site

Before you can start making or editing documents in Flash MX 2004 for a project, you need to define a site that the Project panel can use. In this section, you learn how to define a site and establish a local mirror copy of your site's files on your machine.

1. On your computer, choose a location that you can use to store all of the files with a project. For example, if you're on Windows, you can create a folder named `Sites` at the root of your C drive. If you're on a Mac, you can create a folder named `Sites` at the root of your startup drive, such as Macintosh HD.

2. Inside of the `Sites` folder, copy the `robertreinhardt.com` folder from the `ch03/starter_files` folder located on this book's CD-ROM. As shown in Figure 3-5, the `robertreinhardt.com` folder has two subfolders, `dev` and `wwwroot`.

   The `dev` folder, short for ***dev**elopment*, will contain any source files, specifications, planning documents, raw assets (images, video, and sound), and so on. The `fla` folder inside the `dev` folder holds all Flash documents (.fla files) for the project.

Tip

You could also make an include folder to store ActionScript files (AS files). Feel free to add as many folders in the `dev` folder as you need.

   The `wwwroot` folder will contain any and all files that will be part of the final application, as a publicly accessible Web site or application. All of the Flash movies (.swf files), runtime assets (JPEGs, MP3s, FLVs, and so on), and HTML documents will be kept here. The copy of `wwwroot` from the CD-ROM includes several subfolders, to store external assets necessarily for the Flash movie (.swf file) at runtime.

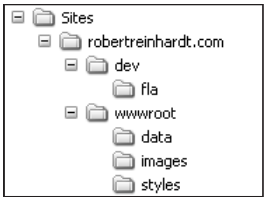
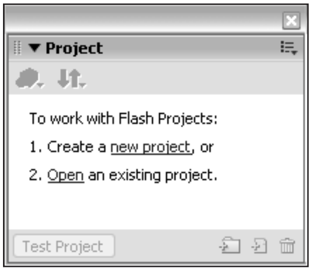


Figure 3-5: The layout of folders for a site named robertreinhardt.com

3. Now you're ready to create a Flash Project file, to put into the project folder you created in the last step. Open Flash MX Professional 2004. Choose Window ➪ Project (Shift+F8).

4. Click the **Create a new project link** in the Project panel (shown in Figure 3-6).

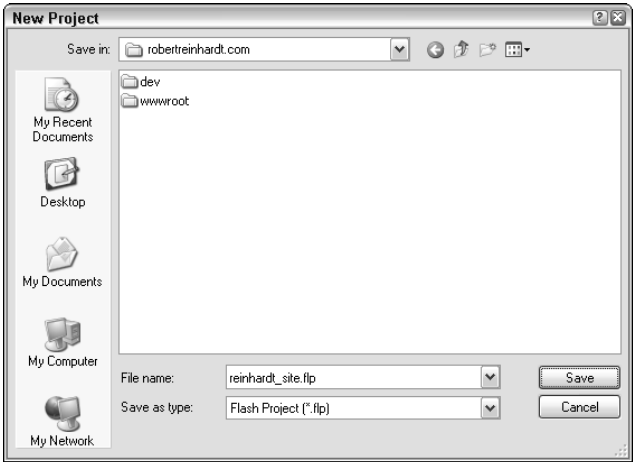


To work with Flash Projects:

1. Create a new project, or
2. Open an existing project.

Figure 3-6: The Project panel

5. In the New Project dialog box, browse to the robertreinhardt.com folder on your computer. Save a new project file named `reinhardt_site.flp` in this location, as shown in Figure 3-7.



File name: reinhardt_site.flp

Save as type: Flash Project (*.flp)

Figure 3-7: The New Project dialog box

6. With a project file created, you're ready to define a site in Flash MX Pro 2004. In the Project panel, click the Version Control button, which features an icon of two arrows pointing in opposite directions. In this menu, choose the Edit Sites option as shown in Figure 3-8.

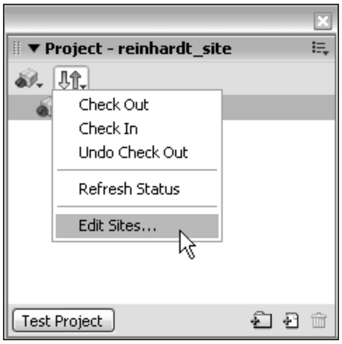
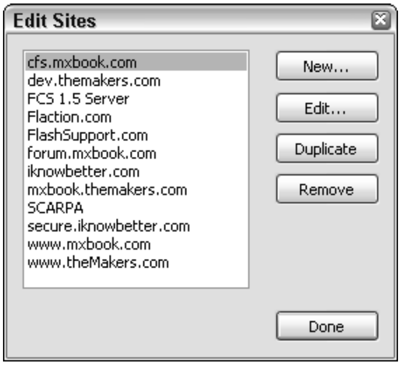


Figure 3-8: The Version Control menu

7. In the Edit Sites dialog box, you may see other sites already defined (as shown in Figure 3-9). If you use Dreamweaver MX or Dreamweaver MX 2004, you can use these sites, or create a new site for the project. For this example, create a new site by clicking the New button.



Edit Sites

cfs.mxbook.com
dev.themakers.com
FCS 1.5 Server
Flaction.com
FlashSupport.com
forum.mxbook.com
iknowbetter.com
mxbook.themakers.com
SCARPA
secure.iknowbetter.com
www.mxbook.com
www.theMakers.com

Figure 3-9: The Edit Sites dialog box

8. In the Site Definition dialog box, specify a name for the site such as robertreinhardt.com. Most importantly, specify the path to the robertreinhardt.com folder in the Local Root field, as shown in Figure 3-10. For the Email and Check Out Name fields, type your own information. In the Connection parameters, you must decide how you will connect to the testing (or live) server that will host the "master" copy of all project documents. You can use a location that's accessible via FTP, the local network, or a SourceSafe database. This location also stores the lock files (.lck files) necessary for members in your team to check in and check out documents. When you are finished specifying the connection details, you may want to click the Test button to make sure that Flash MX Pro 2004 can connect to the location. Click OK to close the dialog box, and click the Done button in the Edit Sites dialog box.

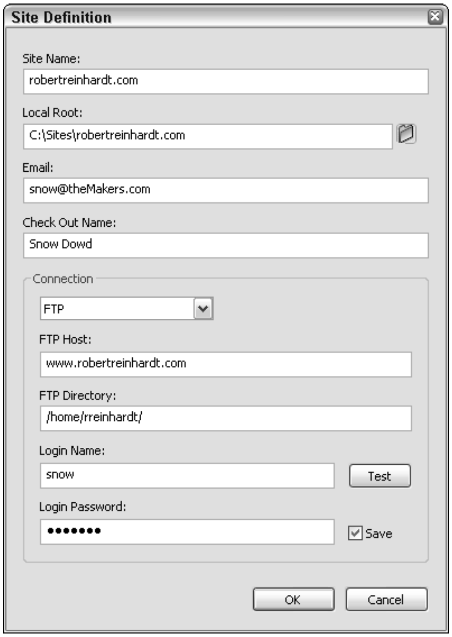
Note

The Connection parameters shown in Figure 3-10 are for demonstration purposes only. These parameters will not connect to an actual FTP site. If you use an FTP connection, make sure that the FTP Directory field specifies the path to the *parent* folder of your public HTML or Web folder for the site. The public Web folder of your server will vary depending on your server's operating system and Web server software.

Tip

It is entirely optional to use the Version Control features of a Flash Project file. If you simply want a way to quickly access all of the documents within a project for yourself, you do not need to create or link a site to your Flash Project file.

9. Now, you will link the newly defined site to your `reinhardt_site.flp` project file. Right-click (or Control+click on Mac) the `reinhardt_site` file in the Project panel, and choose Settings. In the Project Settings dialog box, choose **robertreinhardt.com** in the Site menu. Refer to Figure 3-11. Click OK to close the dialog box.



Site Definition

Site Name: robertreinhardt.com

Local Root: C:\Sites\robertreinhardt.com

Email: snow@theMakers.com

Check Out Name: Snow Dowd

Connection

FTP

FTP Host: www.robertreinhardt.com

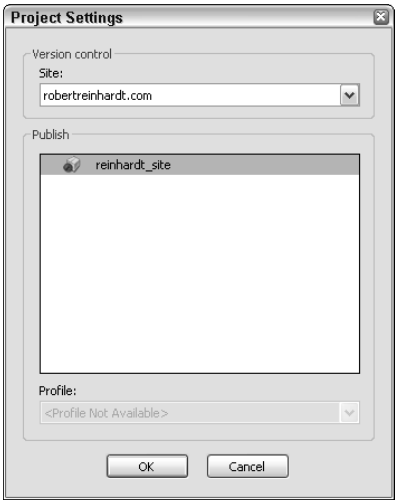
FTP Directory: /home/rreinhardt/

Login Name: snow

Login Password: •••••••

☑ Save

Figure 3-10: The Site Definition dialog box



Project Settings

Version control

Site: robertreinhardt.com

Publish

reinhardt_site

Profile: <Profile Not Available>

Figure 3-11: The Project Settings dialog box

2. Adding files to the project

After you have created a Flash Project file and defined a site for the project, you're ready to start adding files to the project.

1. Begin the process of recreating the folder structure of the local site folders in the `reinhardt_site.flp` project file. In the Project panel, click the Add Folder icon at the lower right corner of the panel. Name the first folder **dev**. Repeat this process until you have created all the folder names you had in the `robertreinhardt.com` folder, including the subfolders. When you are finished, you should have the same folder structure shown in Figure 3-12.

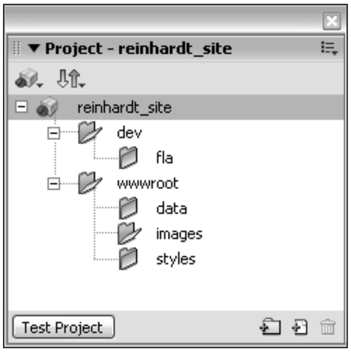


Figure 3-12: The folder structure of the site within the project file

2. Select the `fla` folder you created in the last step, and click the Add File button in the lower right corner of the Project panel. Browse to the Sites ➪ robertreinhardt.com ➪ dev ➪ fla folder and select the `bio_100.fla` file located there. Repeat this process for all of the files contained in the `robertreinhardt.com` folder. Do not add the `reinhardt_site.flp` file itself. When you are finished, your Project panel should resemble Figure 3-13.

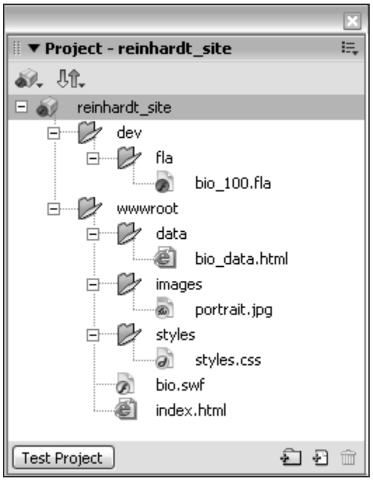


Figure 3-13: The folder and document structure of the site within the project file

3. It is highly likely that you'll have more than one Flash document (.fla file) in a project. As such, you should define the default document for the project. This file should be the master file, the one "most in charge" per se. This could be the Flash document that controls the loading of other runtime assets or the document that contains the most code. In the `dev` folder of the Project panel, right-click (or Control+click on Mac) the `bio_100.fla` document and choose **Make Default Document** in the contextual menu. The icon of the document should change to a downward-pointing green arrow.

3. Committing and editing files in the project

Once you have added files to your project, you should commit the files to your testing server. In this section, you learn how to commit project files and how to open and edit documents from the Project panel.

1. When you are finished creating folders and adding files to your Flash Project file, you should check in the documents to your remote testing server. You can do this procedure only if you have defined a site for the project file. Choose the `reinhardt_site` file at the top of the Project panel. Right-click (or Control+click on the Mac) the filename, and choose **Check In**. Flash MX Pro 2004 then connects to your remote server and checks in the file. When the file has been successfully checked in, a lock appears next to the file (see Figure 3-14).

2. Repeat the process in Step 1 for the `dev` and `wwwroot` folders in the Project panel. When you check in an entire folder, all of the files within the folder will be checked in. When you are finished, you should see locks next to all of your documents, as shown in Figure 3-14.

3. When you're ready to edit a specific file in Flash MX Pro 2004, right-click (or Control+click on Mac) the file in the Project panel and choose **Check Out** in the contextual menu. Try this step with the bio_100.fla document. Once you have checked out this file, double-click the file to edit it in the Flash authoring environment.

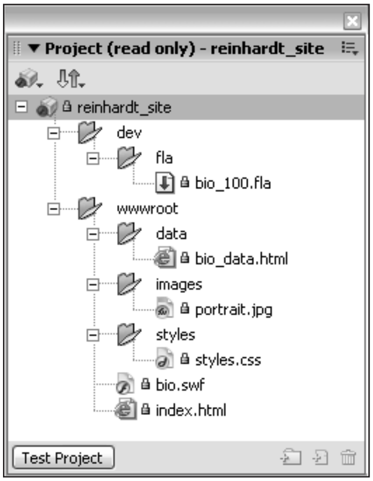
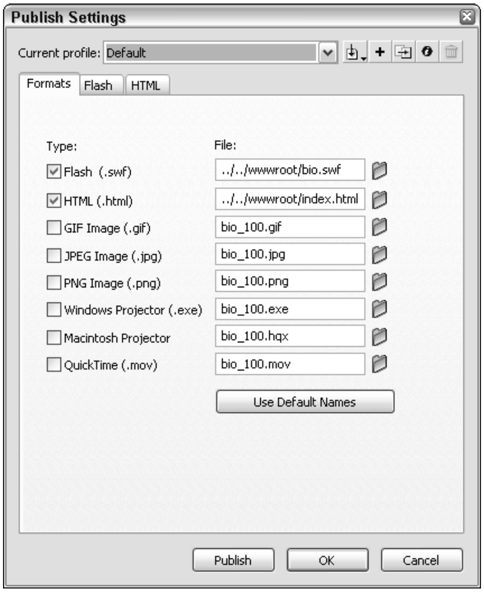


Figure 3-14: A file that is checked in displays a lock icon.

Note When you have checked out a file, you'll see a green check mark next to the file icon. Other members of your team subscribed to the same project will see a lock next to the same file in their Project panels.

4. With the `bio_100.fla` open, let's take a look at how the `bio.swf` file (located in the `wwwroot` folder) is published. Choose File ➪ Publish Settings. In the Formats tab, notice that relative paths are declared for the `bio.swf` and `index.html` files (see Figure 3-15), in the Flash and HTML fields respectively. The `../../wwwroot/` prefix tells Flash MX 2004 to publish these files two folders above the `fla` folder, inside of the `wwwroot` folder. Click Cancel to close the dialog box. Leave the `bio_100.fla` open for the next series of steps in the following section.



Publish Settings

Current profile: Default

Formats | Flash | HTML

Type:	File:
☑ Flash (.swf)	../../wwwroot/bio.swf
☑ HTML (.html)	../../wwwroot/index.html
□ GIF Image (.gif)	bio_100.gif
□ JPEG Image (.jpg)	bio_100.jpg
□ PNG Image (.png)	bio_100.png
□ Windows Projector (.exe)	bio_100.exe
□ Macintosh Projector	bio_100.hqx
□ QuickTime (.mov)	bio_100.mov

Figure 3-15: You can publish files with relative paths in the Formats tab.

4. Adding new files to the project

In this section, you learn how to create a new ActionScript document (.as file), using code extracted from the `bio_100.fla` document. This ActionScript document will be added to the project file as well.

1. From your desktop, browse to the location of the `dev` folder (for example, `C:\Sites\robertreinhardt.com\dev`). At this location, create a new folder named **`includes`**. This folder will be used to store ActionScript files.

2. In Flash MX Professional 2004, add the same folder name (`includes`) as a child of the `dev` folder in the Project panel.

3. Choose File ➪ New. In the General tab of the New Document dialog box, choose ActionScript File and click OK. When the new document opens, save the empty file as **`functions.as`** in the `includes` folder you created in Step 1. Leave the `functions.as` document open.

4. Go back to the bio_100.fla document. In the Timeline window, select frame 1 of the actions layer and open the Actions panel (F9). Select lines 1–48 and press Ctrl+X or ⌘+X to cut the code from the frame.

5. Switch back to the functions.as document, and choose Edit➪Paste (Ctrl+V or ⌘+V) to move the code into the document. Save and close the functions.as document.

6. Now, add the functions.as document to the includes folder of the Project panel. To do this, you'll need to check out the reinhardt_site project file. Then, right-click (Control+click on the Mac) the includes folder and choose Add File. Browse to the includes folder on your local drive, and add the functions.as file. When you are finished, check in the functions.as document and the reinhardt_site project file.

7. Go back to the bio_100.fla document, and select frame 1 of the actions layer and open the Actions panel (F9). Add the following line of code at the top of the Script pane (line 1):

   ```
   #include "../includes/functions.as"
   ```

   This directive tells Flash MX 2004 to insert the contents of the functions.as document at the time of publishing or testing.

8. Save and close the bio_100.fla document. Leave this document checked out while you proceed to the last section.

5. Publishing the entire project

In this final section, you'll learn how to test an entire project, and upload the updated runtime files to your testing server.

1. Before you can publish or test the project, you'll need to unlock the files that will be published by Flash MX 2004: bio.swf and index.html. Check out these files in the wwwroot folder of the Project panel.

2. Click the Test Project button in the lower-left corner of the Project panel. Flash MX Pro 2004 will publish all of the Flash documents (.fla files) in the project file. In our example, there's only one FLA file, bio_100.fla. The newly published bio.swf file, located in the wwwroot folder, then opens in the Test Movie environment.

3. If everything is working correctly, check in the bio.swf, index.html, and bio_100.fla documents. If there was an error, double-check the code added in the last section for syntax errors. The Output panel will likely provide clues about any errors associated with improper URLs for runtime assets.

On the CD-ROM

You can find the final site files in the ch03/final_files folder of the *Flash MX 2004 Bible* CD-ROM. If you try to open the reinhardt_site.flp document from the CD-ROM (or a copy of it), you'll likely need to relink the documents to the locations of the copies on your system.

Web Resource

We'd like to know what you thought about this chapter. Visit www.flashsupport.com/feedback to fill out an online form with your comments.

Summary

✦ Your clients rely on you to understand and guide the production process involved with Flash content creation.

✦ Careful planning helps you to create Flash solutions that best meet the goals of your project. The technical issues, such as usability, target audience, and delivery platform, should be balanced with the aesthetic aspects of experience design.

✦ To structure the development of Flash projects many Web developers use a two-phase production model that involves six milestones: Business Initiative, Creative Solutions, Approval, Production, QA, and Delivery.

✦ During the production period, it is helpful to keep six key concepts in mind: asset assembly, a master Flash architecture, a local test environment, HTML page layout, a server staging environment, and proper QA testing. After production is finished, you also need to devise a strategy for systematic maintenance.

✦ You can use the Project panel in Flash MX Professional 2004 to manage all of the Flash documents for a site or application.

✦ ✦ ✦