

Planning Your Project and Setting Up Dreamweaver

Although it might be more exhilarating to start designing web pages on the first day of your project, experience has taught us that a successful web development project starts with some strategy. First you want to take a step back and look at the big picture. What is your site or application going to do? Who's going to use it? How does it have to work? How should it be organized?

Next, to make the most of Dreamweaver, take a little time to understand how the software creates, interprets, and displays web pages. Get the software set up in a way that suits your workflow, and then finally start developing your site.

Even then, before you get into the process of building actual pages, you're going to need to spend some time setting up your site in Dreamweaver, and creating the shared templates, library items, and other assets that will enable a rapid, efficient development process.

CHAPTER 1 ■ Planning and Preparing for a Dreamweaver Project

CHAPTER 2 ■ Web Pages Deconstructed

CHAPTER 3 ■ Setting Up Your Workspace and Your Site

CHAPTER 4 ■ Saving Labor with Templates and Libraries

Planning and Preparing for a Dreamweaver Project

Dreamweaver can help your project run more smoothly whether you are building a website from scratch all by yourself for yourself, collaborating with a multidisciplinary team to deliver a web application for a client, or anything in between. Regardless of the scope of your project, take some time in advance to think through the *architecture* of your site (fundamentally, the layout structure of the pages as a whole, and the options for navigating the site), develop a look-and-feel (graphic design and interface), and gather the *content* (the information you want to include on the site pages). When this preliminary work is done, you can plunge into the development, staging, launching, and maintenance of the site.

This chapter assists you in sorting through the elements of your preferred methodology before you get down to brass tacks with Dreamweaver. Topics addressed in this chapter include the following:

- **When to use Dreamweaver**
- **Nailing down your process**
- **Gathering requirements for your web project**
- **Designing the project's information architecture**

When to Use Dreamweaver

As the Web has grown and developed, so has Dreamweaver. Dreamweaver has evolved from an HTML editor to a full-featured web production and application development tool. Each version introduces new features that make the web development process easier and faster whether you are involved in the front end (design and page production) or the back end (scripting and application development).

Dreamweaver goes beyond enabling you to design and create web pages—it helps you manage your entire site from the top down. Best of all, it enables a team of people to work together on the same project. If you are working with collaborators, chances are not everyone is working in Dreamweaver. That's fine. Dreamweaver produces clean code that even the most hardcore code jockey can't complain about.

With Dreamweaver, though, you can keep your part of the project all in one place, and you can do all of your work inside the application (at least after you're done doodling on cocktail napkins).

Using Dreamweaver with Other Applications

Dreamweaver MX 2004 is an incredibly powerful web development tool, but of course it can't do every single thing a designer or developer would like to do. Macromedia wisely provides tight integration with its other popular web applications: Fireworks, Flash, HomeSite+, and ColdFusion. You can also use files from Macromedia Director, Microsoft Word and Excel, and other graphics programs in your Dreamweaver pages.

Fireworks and Other Graphics Software

Fireworks is a unique tool expressly designed for the development and optimization of web graphics. Fireworks is where you can develop and refine the graphical look-and-feel for the site—the site's logo, the graphics, the navigational elements, and so on. The entire site won't live inside Fireworks the way it will in Dreamweaver, but because they're both Macromedia products designed to work together, any graphics you develop or import into Fireworks will flow easily into your site templates and pages over on the Dreamweaver side.

For more information on using Fireworks and other graphics programs, see the CD Bonus Chapter, "Slicing, Optimizing, and Exporting Images," and Chapter 13, "Designing Navigation Objects."

You don't have to use Fireworks to create and develop your site graphics, however. Dreamweaver accepts web graphic files (GIFs, JPEGs, and PNGs) from any graphics software program including Adobe Photoshop, Adobe Illustrator, and Jasc Paint Shop Pro.

In Dreamweaver preferences (Edit → Preferences / Dreamweaver → Preferences) you can select external editors for specific file types. For example, you can choose to have Dreamweaver open Photoshop when you want to edit a JPEG image. To open the external editor, select the image and click on the Edit icon in the Property inspector. For more details on using the Property inspector, see Chapter 3, “Setting Up Your Workspace and Your Site.”

However, there are two major advantages to using Fireworks with Dreamweaver:

- Fireworks is specifically designed for creating web graphics and includes a multitude of special features for doing so.
- Fireworks is integrated with Dreamweaver and can export both HTML code and graphic files directly to Dreamweaver.

Flash

Flash is the web designer’s favorite tool for developing interactive movies, animations, and every other kind of beast that slithers, crawls, runs, or flies across your screen. Flash movies are optimized for streaming over the Web, and widely accepted as a format. The newest version of Flash includes a professional version that features extensive video support as well as Flash application development. Flash is the first choice when you want your users to be able to, say, play a video game at your website. Artists love Flash too (see www.snarg.net/ for a hypnotic example of what we’re talking about).

You can create Flash text and Flash buttons in Dreamweaver (whether or not you have Flash installed on your computer). Dreamweaver MX 2004 also includes support for *Flash elements*—prebuilt Flash components that can be downloaded and added to your pages.

Flash is integrated with Dreamweaver. You can export Flash files to your Dreamweaver site, insert Flash movies in your web pages, and use Dreamweaver to set playback and display options. If you have Flash installed on your computer, you can also use Dreamweaver to update the links in Flash movies or launch and edit Flash files from Dreamweaver.

For more details on working with Flash, see Chapter 11, “Getting into Flash.”

Director

Director is a multimedia authoring tool used to create interactive multimedia files, including web content (Shockwave files), multimedia CDs and DVDs, games, movies, and cartoons. Files created in Director can include photo-quality images, digital video, audio, 2D and 3D animation, and even Flash content. Director files can be deployed to multiple environments, including the Web, kiosks, CDs, and DVDs. Chances are, last time you stuck a

CD in your drive and watched a little promo or clicked on a bulbous shiny set of interface buttons, they were developed in Director—this book’s CD features a Director interface.

Macromedia statistics (www.macromedia.com/software/player_census/shockwaveplayer/) show that 375 million web users have the Shockwave Player. Although Director is not actually integrated with Dreamweaver, you can use Dreamweaver to insert Shockwave files in your pages, and set playback and display options.

For more information on Director, see Chapter 10, “Adding Graphics and Multimedia.”

Getting Your Process Squared Away

Before you fire up the software and start cranking out web pages, take a step back to sort out your process (methodology). Nowadays, most web design and development projects are collaborative and require a lot of coordination among team members. Yes, it’s true that if you’re running a one-person project or shop, you don’t have to answer to anybody, you don’t have to use anyone else’s lingo, you don’t have to adhere to anyone’s deadlines, and no one is going to second-guess your work. But even then you’re going to have to figure out what to do first, what part of the project depends on other parts being completed first (sometimes referred to, for short, as *dependencies*), and what your timeline and milestones are going to need to be.

In most situations, you’ve got a “someone” to answer to, whether it’s your boss, your client, or simply your audience. That’s right, web design and development requires you to anticipate and meet your audience’s needs if you expect them to come to your site, use your interactive application, register with your enterprise, or come back again after the first visit. Furthermore, in most commercial projects, you’re going to have to collaborate with somebody, or with a whole team of somebodies. There might be a branding expert, a writer (perhaps called a content developer), some developers (technical architects, front-end scripters, back-end coders, middleware specialists, and so on), and possibly a project manager. Oh, yes, and a visual designer or graphic designer.

You may be working with people who cut their teeth in the field of professional services, interactive or advertising agencies, publishing, and software development. You’re going to discover that everyone has a different name for the same thing (is it a *storyboard* or a *wireframe*, a *site map* or *thumbnail series*, *use cases* or *process flows*?), and most people see the project revolving around their discipline. In any collaborative project, some time—at least an hour—should be spent up front hashing out the division of labor, the dependencies (such as, “I can’t develop the content inventory until you finish the site map”), the points of handoff or turnover, and the milestones and deliverables expected by the client (even if the client is just your boss).

For more ideas and discussion about various web-development methodologies (and there are a number of equally valid approaches), check out the author-created website for this book, at <http://dreamweaversavvy.com/>.

Gathering Requirements

How are you going to know what to put into your site or application unless you spend some time and effort learning the needs of your website's or application's eventual users? (Consultants call this stage of a project *discovery*—not to be confused with lawyers pawing through your files.) This discovery phase should involve interviewing representatives of every audience type or anyone else with a stake in the usefulness and success of the site. This means not just your boss or client, your client's boss or team members, and other obvious stakeholders, but also, if at all possible, some potential users of the site—often customers, partners, or vendors. Find out what they want. Your role is to meet the needs of both the client and the users. Clients do not always know what their users want as well as those clients think they do. Also, it makes a killer argument when the client has gotten attached to some horrible idea to be able to say, “But your site's users don't want that. See, here in these interview notes, they say they'd never come back if you had *that* as part of your site.” It's also worth taking the time to check out the competition—look at their sites to get an idea about the features they offer, and what works and what doesn't.

This leads to the first commandment of web design (perhaps the only commandment—we're not sure):

Know Thy Audience

What if they built a website and nobody came? They did. And nobody did. It was called the dotcom bubble. Just because you can sell your boss, or your client, or a venture capitalist (VC) on an idea doesn't mean that people are going to come and pay you to keep executing that idea. Understand your audience. Go meet them if possible. Interview them, but also watch them as they work. Study what they like and dislike. Learn as much as you can about usability. If your ultimate product isn't useable, guess what? People won't use it. If your site doesn't meet a need, then no one will need it.

So think long and hard about the needs and desires of your audience. What are they reaching out for? How can you satisfy those needs? Get that straight and the rest of the project will practically take care of itself.

Once your site is up and running online, be sure to follow site statistics and usage patterns. See what your traffic actually is, and which parts of your site get visited and which parts are ignored. Pay attention to the feedback you receive from visitors. Good websites change over time to address the current needs of their audience.

Getting the Information Architecture Right

Information architecture is a \$10 word that means how your site's information is organized. What do people see first when they come in the front door? How many levels down is certain information buried? How many clicks does it take to get to crucial information? What is the structure of the navigation? Dreamweaver won't figure any of this out for you. Sure, once you've sorted it out, Dreamweaver is an awesome tool for maintaining the site map, navigation links, and so on. But you have to do the hard thinking first.

Fortunately, because Dreamweaver is so flexible and easy to use, you can make mistakes when you start and still correct them later on in the project. That's right, no matter how carefully you gather your requirements, know thy audience, or massage your client, guess what? New requirements will emerge at the 11th hour. Projects you've never heard of will suddenly demand to be integrated into your pristine site map. With a few points and clicks in Dreamweaver, subsections can be promoted to top-level categories, and entire site areas can be snipped out and placed in limbo. I'd like to see a "real" architect try to rearrange a real building once the contractors are in the house!

Still, just because things are inevitably going to change, that's no excuse for not trying to get it right at first. In fact, getting your architecture clarified at the beginning makes it that much easier to *track the changes* as they emerge. Think about it. It's a lot easier to see what's changed if you know what it changed from!

Also, remember those other people on the team. They need to know where you're planning on putting stuff. They may be writing scripts and have to know what directory (a.k.a. folder) a certain piece of content is going to live in. They may need to know how many levels down their application is going to be running. Sorting out the site's architecture is step one of designing any substantial web project.

ENSURING THAT YOUR SITE IS ACCESSIBLE

One of the most important changes to the specifications for HTML in Version 4 is the inclusion of requirements that your site be accessible to people with disabilities. This means that your work can be used and appreciated by the estimated 10 percent of the world's population with some sort of physical disability that makes it difficult to access most websites. The World Wide Web Consortium (W3C) has set up a special organization and website that provide helpful checklists, guidelines, and ideas in support of the Web Accessibility Initiative (WAI). Check out www.w3.org/WAI/ to find out how to insert accessibility into your website relatively painlessly. Another very useful site that serves as both an example of a well designed, accessible website and as a tutorial is the Web Accessibility In Mind (WebAIM) site (www.webaim.org/). For information about using Dreamweaver's accessibility features, see Chapter 33, "Using Dreamweaver to Make Your Site Accessible."

Developing a Site Map

A great way to get an overview of a website at a glance is to create and maintain a site map. Ultimately, Dreamweaver can generate or show you a site-map type view of your site, but that's after you've actually created all the pages. When you're just getting started, you can literally just draw it by hand or use any illustration software to put it together. A site map looks something like a family tree. Pages are represented as boxes and labeled. You can indicate the name of a page or—if you really want to get into it at this stage—what the page will contain. Navigational links are represented as lines between boxes. Then, like a family tree, child and sibling pages all stem from a parent page, and you can take this organization down as many levels as you like. Some people make the boxes smaller as they go down in levels. Often the third level of links is represented as a simple text list. The level of detail depends on your own needs or those of your client.

Figure 1.1 shows a sample site map for an e-commerce site.

When you are developing an interactive application (as opposed to a series of static, linked pages), the site map can also be referred to as a *process flow*, because it shows how the user might flow through the various pages or screens to accomplish some task. For example, let's say you developed a new search engine for a site. You could create a process-flow diagram showing how users can go from the basic search box to the advanced search page or to the search tips, and then ultimately to the search results.

You can create whatever codes or symbols you'd like to indicate different types of pages. Use dotted lines for dynamic pages (pages that are created from a database), or use rounded edges for new pages. Have fun with it. This is information design without that tedious “making it work” part.

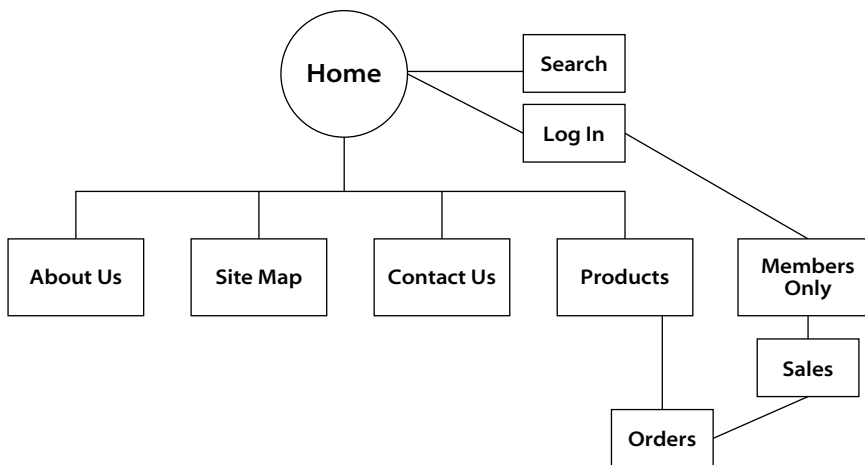


Figure 1.1

This e-commerce site has a main page with links for Search and Log In as well as links to four subsections reached from the main page. The Log In page links to another subsection for members only.

Creating Wireframes and Storyboards

If you are developing a large site with several subsections, you may want to create *wireframes* for the project. The name *wireframe* comes from the skeletal wires that underlie a piece of sculpture. A wireframe is a simple page of text that includes the purpose of one specific web page and all entry and exit points from the page—that's it! Wireframes do not include design or content elements.

Once you have created wireframes for every page in the site, and made final decisions about the overall site structure, you can then create *storyboards* for each page that show the content, layout, navigation elements, design elements and other graphics.

Figure 1.2 shows a mocked-up wireframe page for the home page in our imaginary e-commerce site.

For more information on using wireframes and storyboards for web projects, see www.grokdotcom.com/wireframing.htm.

Ready, Set, Rumble!

This chapter explained some of the planning and organizational work you'll need to do before you launch a serious web-development project. This includes determining when it's appropriate to use Dreamweaver, Fireworks, Flash, and other software; hashing out a process or methodology for developing your site or application; figuring out how to collaborate with other team members (if necessary); gathering requirements for your site; and developing an information architecture.

Figure 1.2

This simple wireframe shows the purpose of the page as well as all links.

Home Page

Purpose: main entry page for e-commerce site

Links to:

- Search
- Log In
- About Us
- Site Map
- Contact Us
- Products

This information is useful no matter what tools you use to assist you as a web designer and developer. Don't ignore these steps because they involve thought processes and decisions that your software applications can't do for you. Once you know how you're going to design your project, what you're going to do it with, how and when you are going to use your tools, who the product is for, and how the information at the site will be organized, you're ready to start cranking away in Dreamweaver. The next chapter will take you through the elements of a web page and a website and will even show you some actual Dreamweaver screens! Remember, have fun.

