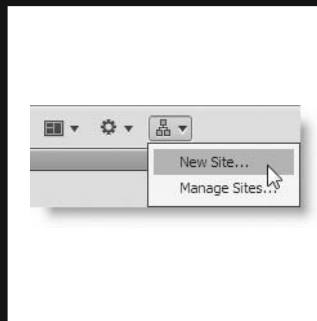
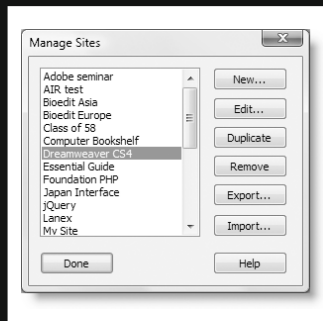


2 SETTING UP A SITE IN DREAMWEAVER



Dreamweaver is not just a tool for creating web pages; it's for creating websites. You create an exact copy of the website on your local computer, tell Dreamweaver how to connect with the remote web server, and upload it. When pages need updating, you download the latest version from the remote server (unless you work on your own and know the remote version hasn't changed), make any amendments, and upload it again. Even if you don't yet have a remote server, Dreamweaver expects you to organize your files exactly as they would be in a real site. This is necessary because Dreamweaver needs to keep track of the images and other assets, such as style sheets and JavaScript files that make up each page. If you decide to move files to a different location, Dreamweaver automatically updates internal links, but it can't do so unless it knows certain basic details about the site. So, defining your site within Dreamweaver is an essential first step before you even think about creating a web page.

In this chapter, you'll learn about the following:

- Choosing the best location for your files
- Understanding the difference between document- and root-relative links
- Creating virtual hosts on Windows and Mac OS X
- Setting up and testing a Dreamweaver site definition
- Integrating a site with a Subversion repository
- Backing up and migrating your site definitions

This is quite a long chapter, but many parts of it are relevant only to some readers. For instance, you can skip the entire section on virtual hosts if you decide not to use them. Also, in many cases, there are separate sets of instructions for Windows and Mac. However, don't be in a rush to get through this chapter. Site definition in Dreamweaver is not difficult, but you need to get it right, particularly when working with a server-side technology like PHP—the focus of the second half of this book.

Dreamweaver supports three popular server-side technologies: ASP (sometimes referred to as classic ASP), ColdFusion, and PHP. In this book, I cover only PHP, but setting up a site for any server-side technology involves the same process. Server-side pages need to be processed (or **parsed**) before you can see the dynamically generated output. This affects how you set up your site in Dreamweaver. So, let's begin with the considerations for site organization.

Deciding how to organize your site

Site definition in Dreamweaver involves the following three basic steps:

1. Telling Dreamweaver where to find the version of the site on your local computer.
2. Registering details of the remote server. This is the live version of the site on the Internet.
3. Registering details of the server that will process server-side code (in the case of this book, PHP) while testing pages locally.

If you don't yet have a remote server, you can skip step 2 and register the details later. Likewise, if you don't plan to use any server-side code in the site, you can omit step 3. Even if you don't have any immediate plans to use PHP, I suggest that you read through the next few pages so that you don't make any decisions you might later regret. Planning ahead saves time later.

Deciding where to test your pages

2

Building PHP pages involves a lot of testing—much more than you might normally do with a static website. It's not only a question of what your pages look like; you also need to check that the dynamic code is working as expected. Dreamweaver doesn't care where your testing server is, as long as it knows where to find it, there's an available connection, and, of course, the server is capable of handling PHP pages. This means that you can test on your local machine, another computer on a local network, or a remote host.

These are the advantages of creating a local test environment:

- **Security:** Dreamweaver installs files on the testing server to communicate with your database. If you use your remote server for testing, a malicious attacker might be able to use these files to gain access to your data. As long as your computer is protected by a firewall, this danger is eliminated by testing locally.
- **Stability:** If an error in your code causes the server to slow down or even crash, the only person affected is you. Keep your mistakes to yourself; don't inflict them on others. You're likely to make a lot of mistakes until you gain a level of competence in PHP (don't let that put you off—it's an important part of the learning process).
- **Ease of organization:** You can build the website offline, test it, and upload it directly to the live site when it's ready. If you use your remote server for testing, the test site is visible to everyone on the Internet unless you build it in a separate testing folder or password protect it. The first option involves changing all internal links as soon as the site goes live. Password protection also affects you, so neither option is optimal.
- **Speed:** There's no waiting. Even with a broadband connection, the response is usually slower from a remote server.
- **Convenience:** You can continue work even if there is a disruption to your Internet service.

There are also disadvantages to creating a local test environment:

- **Setup time:** Each piece of software requires a multimegabyte download, which then must be set up and configured.
- **Complexity:** Some people find configuring the software daunting. However, if setting up a testing environment is beyond your capabilities, you should perhaps reconsider whether you're ready to work with a server-side technology. PHP is not difficult, but it does require clear thought and concentration.

In previous books, I have given detailed instructions about how to set up a local testing environment with a web server, PHP, the MySQL database, and a graphical interface to MySQL called phpMyAdmin. However, new versions of PHP, MySQL, and phpMyAdmin are expected to be released around the time this book is published, so anything I write risks being out of date by the time you read it.

If you don't already have a PHP testing environment on your local computer, I suggest that you get XAMPP for Windows (<http://www.apachefriends.org/en/xampp-windows.html>) or MAMP for Mac OS X (<http://www.mamp.info/en/mamp.html>). Both give you a fully working PHP/MySQL testing environment on your local computer. They're easy to install, and should have you up and running in a few minutes. Both XAMPP and MAMP have online forums, where you should be able to get help if you run into problems.

If you already have Microsoft Internet Information Services (IIS) installed for development with ASP or ASP.NET, you can also install PHP to run with IIS. The official IIS website gives instructions for installing PHP on Vista: <http://learn.iis.net/page.aspx/246/using-fastcgi-to-host-php-applications-on-iis7/>. Windows XP uses a different version of IIS, but a web search for "install PHP on IIS 6" should provide some helpful guides. However, don't use IIS unless you also need ASP or ASP.NET, as some parts of PHP are not supported on IIS.

Although Chapters 3 through 9 of this book don't require PHP, I recommend that you set up a PHP testing environment at this stage rather than wait till later.

Chapter 9 covers the creation of an online form. The form is built entirely with HTML, but processing an online form requires server-side scripting. Chapter 11 shows you how to do this with PHP. However, you can use any other mail-processing script with the form from Chapter 9.

Before setting up a site in Dreamweaver, you need to make a number of decisions. Figure 2-1 summarizes the process, which is described in detail in the following sections.

Choosing the appropriate file name extension

Web servers use the three- or four-letter extension at the end of a web page's file name to decide how to display it. Generally speaking, files that end in `.html` or `.htm` can contain only static HTML, JavaScript, and CSS. For PHP server-side code to be processed, you need to give the file names a `.php` extension.

You may be wondering what the difference is between `.html` and `.htm`. There isn't any. The use of `.htm` dates back to the days when Windows file names were limited to an 8.3 format (a maximum of eight characters followed by a period and a three-character file name extension). All operating systems now support `.html`, which is the default file name extension that Dreamweaver uses for static web pages.

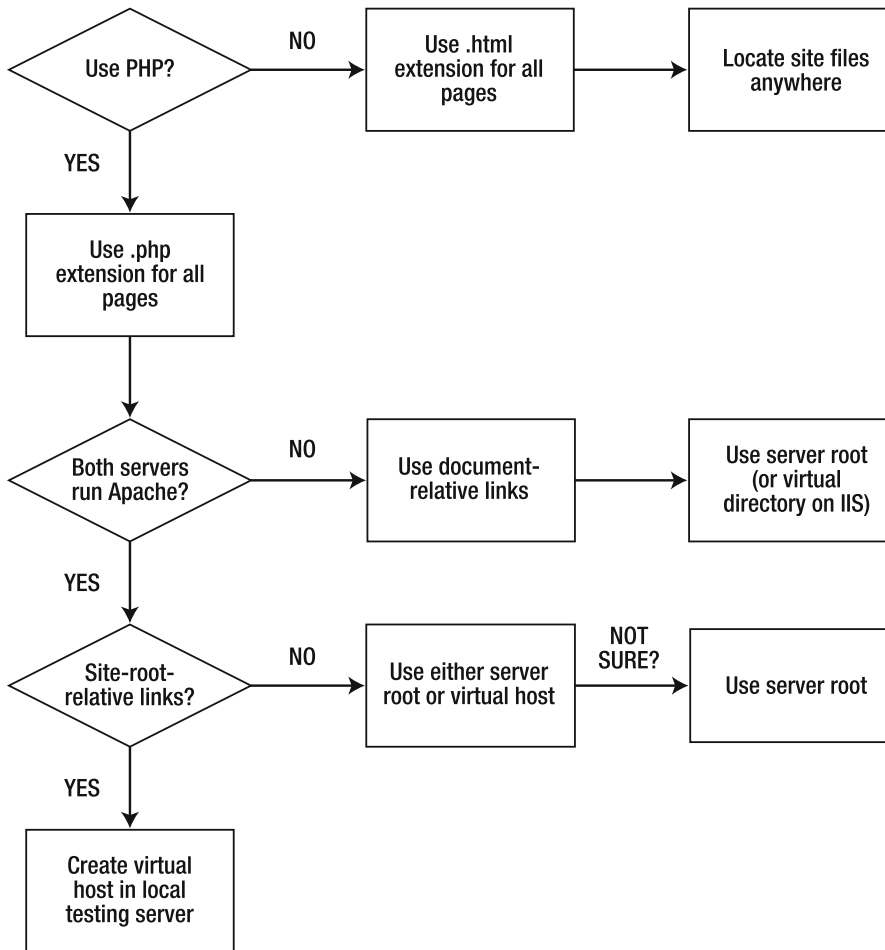


Figure 2-1. Deciding where to locate your Dreamweaver site on a local testing server

As with all rules, there are exceptions. It is possible to configure a server to process PHP even if it's in an `.html` or `.htm` file. However, it's not recommended, and it makes working with PHP in Dreamweaver more difficult. You might also come across alternative file name extensions for PHP, such as `.php4` or `.php5`. Although Dreamweaver recognizes both of these alternatives, you need to edit the file name extension manually when creating files, so use `.php` exclusively unless you have a pressing reason to do otherwise.

It's perfectly acceptable to mix `.html`, `.htm`, and `.php` files in the same website. However, when planning a new website, it's a good idea to use the same file name extension for all pages. That means that if you plan to use PHP anywhere in your site, all pages should have a `.php` file name extension, even if some pages don't contain any PHP code. Technically speaking, the web server sends every page that has a `.php` file name extension to the PHP engine for processing, so it's a wasted journey if the page contains only static HTML. In practice, the difference in the time it takes to serve the page is infinitesimal. The

advantage of using a .php file name extension is that you can add server-side code to the page at any time in the future without changing the URL.

In summary, if the server where your website will be hosted is capable of processing PHP, use a .php file name extension for every page.

In spite of my recommendation to use a .php file name extension for all pages, all the examples in Chapters 3 through 8 use .html. I have done this so that readers who don't have access to a PHP server can use the files that are available for download.

While on the subject of file names, let's lay down some basic rules for naming files and folders on a website:

- Never use spaces in the names of files or folders. Replace spaces in existing names with an underscore (_) or hyphen (-). Spaces in file names and folders are likely to cause broken links on many servers.
- Most web servers are case-sensitive. Avoid problems by using all lowercase for names.
- Many web browsers can handle Internationalized Resource Identifiers (IRIs), which accept file and folder names that use accented characters and different writing systems, such as Chinese and Japanese (see <http://www.w3.org/International/articles/idn-and-iri/>). However, to be safe, you should stick to unaccented letters, numbers, underscores, and hyphens. Do not use symbols, such as ?, #, or /.

Choosing document- or root-relative links

By default, Dreamweaver creates internal links that are relative to the current document, but it also offers the option to use links relative to the site root. With a static website, it doesn't matter which you choose, but with a PHP site, the decision isn't quite so simple. If you're not sure what the difference is, the following explanation should help.

Understanding the difference

Let's say you have a simple website structure like that shown in Figure 2-2.

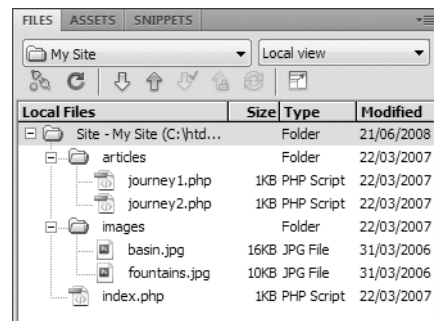


Figure 2-2.
A simple website structure displayed in the Dreamweaver Files panel

If `index.php` contains a link to `journey1.php`, Dreamweaver creates code that looks like this:

```
<a href="articles/journey1.php">Read more</a>
```

And a link back to `index.php` inside `journey1.php` looks like this:

```
<a href="../index.php">Back to main page</a>
```

Similarly, if `index.php` contains the image called `fountains.jpg`, the `` tag looks like this (I have omitted all attributes other than `src`, because that's the only one we're interested in at the moment):

```

```

A reference to the same image in `journey1.php`, however, looks like this:

```

```

The `../` before `index.php` and the `images` folder name tells the web server that it needs to look one level higher in the website hierarchy to find the correct folder. If you change the structure of the website using the Files panel, Dreamweaver automatically updates all links, adding or removing the requisite number of `../` to ensure that everything works as intended.

However, many developers prefer to make the links relative to the site root, rather than the document. With root-relative links, the two links look like this:

```
<a href="/articles/journey1.php">Read more</a>
<a href="/index.php">Back to main page</a>
```

The `` tag in *both* `index.php` and `journey1.php` looks like this:

```

```

The difference is that the pathname always begins with a leading forward slash, which indicates the top level of the site—in other words, the site root.

Which should you choose?

You might wonder why the document-relative vs. root-relative issue matters. After all, both achieve the same thing. When building static sites with `.html` pages, it doesn't make any difference which you choose.

However, root-relative links can be extremely useful with PHP. The advantage is that the link to the image is identical in `index.php` and `journey1.php`, even though the pages are at different levels of the site hierarchy. This means that you can put some of your code, such as a navigation menu, in an external file and the links will always work. As you'll see in Chapter 12, your site navigation menu can be included in multiple pages using a simple PHP command, and changes to the external file are automatically propagated to all of them—a great time-saver.

Because root-relative links are so useful, you may think that they're the best choice for a PHP site. Unfortunately, life is not quite so simple. Although links relative to the site root are essential *inside* external files, PHP expects you to include the external file using a document-relative link. Moreover, if your hosting company runs PHP on a Windows server using IIS, choosing root-relative links in Dreamweaver generates PHP code that results in errors galore and a site that doesn't work.

If you're new to PHP, all this might sound confusing. Don't worry about the details for the moment (if you're curious, the explanation is in Chapter 12). For development with PHP, I recommend the following:

- In the vast majority of cases, choose the Dreamweaver default: links relative to the document. When creating external files that will be included in other pages, you can easily override this and create individual links relative to the site root.
- Choose links relative to the site root only if both your testing environment and remote server run on the Apache web server. If you choose root-relative links, you need to set up a virtual host for each site in your local testing environment.

Choosing where to store your files

With a static website consisting of .html pages, it doesn't matter where you locate files on your local computer. You simply tell Dreamweaver which folder is the site root, and that's it. With a PHP site, though, Dreamweaver needs to send the files to the testing server before it can display the output. This leaves you with the following options:

- Store the files with your ordinary documents and set up the testing server in a different location. Unless the testing server is on a different computer, this is a waste of disk space, as you end up with two copies of every file.
- Store the files in a subfolder of the server root of your testing server. This is the best way to set up a site that uses document-relative links.
- Create a virtual host on your testing server and store the files there. You must do this if you use site-root-relative links. This setup is also suitable for document-relative links.
- If you don't want to set up a local testing environment, it doesn't matter where you store your files locally. However, I cannot emphasize strongly enough that using a remote server as the testing server is *not* recommended.

Finding the testing server root

The location of your testing server root depends on your operating system and how you installed the testing environment.

Apache on Windows

The Apache server root is a folder called htdocs. If you installed Apache 2.2 as a separate program, the server root is at the following location:

```
C:\Program Files\Apache Software Foundation\Apache2.2\htdocs
```


I don't think it's a good idea to store your web files in among all your program files, so I suggest that you move the server root to a different location, as follows:

1. Create a new folder called `htdocs`. The actual location is not important, but I find it best to put it at the top level of my C drive.
2. You need to edit the Apache configuration file, `httpd.conf`, which is located at `C:\Program Files\Apache Software Foundation\Apache2.2\conf\httpd.conf`.
 - In Windows XP, you can open the file directly in Notepad or any other text editor by double-clicking its icon in Windows Explorer.
 - In Vista, you need to select Run as administrator even if you are logged in to an administrator account. For Notepad, go to Start ► All Programs ► Accessories, right-click Notepad, and select Run as administrator from the context menu. Enter your administrator password when prompted. Inside Notepad, select File ► Open and navigate to `httpd.conf`. The Open dialog box in Notepad shows only `.txt` files, so you need to select All Files (*.*) from the drop-down menu at the bottom right of the dialog box.
3. Locate the following section in `httpd.conf`:

```
144 #
145 # DocumentRoot: The directory out of which you will serve your
146 # documents. By default, all requests are taken from this directory, but
147 # symbolic links and aliases may be used to point to other locations.
148 #
149 DocumentRoot "C:/Program Files/Apache Software Foundation/Apache2.2/htdocs"
150
```

4. Change the pathname shown on line 149 of the preceding screenshot to the same as your new folder. (Use the line numbers simply as a guide; they are not part of the file and may be different in a later version of Apache). In my case, I change it to this:

```
DocumentRoot "C:/htdocs"
```

Make sure that you use forward slashes in the pathname in steps 4 and 5, instead of using the Windows convention of backward slashes.

5. Scroll down about 30 lines until you find this section:

```
174 #
175 # This should be changed to whatever you set DocumentRoot to.
176 #
177 <Directory "C:/Program Files/Apache Software Foundation/Apache2.2/htdocs">
```

The instruction shown on line 175 is pretty straightforward: change the pathname to match the previous change. In my case, I end up with this:

```
<Directory "C:/htdocs">
```

6. Save `httpd.conf` and restart Apache for the changes to take effect.

XAMPP on Windows

The server root should be at the following location:

```
C:\xampp\htdocs
```

This location is fine; you don't need to move it.

IIS on Windows

The Default Web Site server root on both Windows Vista and XP is located at this location:

```
C:\Inetpub\wwwroot
```

On Windows Vista, you can define as many websites as you like in the Internet Information Services (IIS) Manager. Right-click Sites in the Connections panel on the left, and select Add Web Site from the context menu. The server root is the location specified in the Physical path field of Content Directory.

Apache on Mac OS X

If you are using the version of Apache that comes preinstalled in Mac OS X, you have two choices of server root. The main one is located here:

```
Macintosh HD:Library:WebServer:Documents
```

Every user account on a Mac also has its own dedicated server root, located here:

```
Macintosh HD:Users:username:Sites
```

Any site within this folder can be viewed in a browser using the address `http://localhost/~username/` followed by the name of the site's subfolder, where *username* is the name of your home folder. The address for the main server root is simply `http://localhost/`, so it is probably the most convenient to use, unless you share the computer with others and want to keep things separate.

MAMP on Mac OS X

The server root is at the following location:

```
Macintosh HD:Users:username:Applications:MAMP:htdocs
```

MAMP installs Apache, PHP, MySQL, and phpMyAdmin as a complete package. This poses a potential conflict, because Mac OS X comes with a preinstalled version of Apache, and you can't normally run both of them simultaneously. To get around this, MAMP uses nonstandard settings to serve web pages and communicate with the MySQL database. Everything in MAMP is configured to work together smoothly. So, if you're using MAMP, use the MAMP version of everything. To avoid problems when you deploy your site on your remote server, turn off any conflicting programs and change the MAMP settings as follows:

1. Make sure that the preinstalled version of Apache is not running by opening System Preferences ► Internet & Network ► Sharing. Select Web Sharing from the list of Services and make sure there is no check mark in the checkbox alongside. The panel on the right should display Web Sharing: Off.

2. If you have ever installed MySQL on your computer, stop the MySQL server and change any system preferences that start it automatically whenever your computer starts up. MySQL is not preinstalled on Mac OS X, so there's nothing to worry about if you have never installed it.
3. Click the Preferences button on the MAMP control panel. In the dialog box that opens, select Ports, and click the button labeled Set to default Apache and MySQL ports. The values in Apache Port and MySQL Port should be the same as those shown in Figure 2-3.

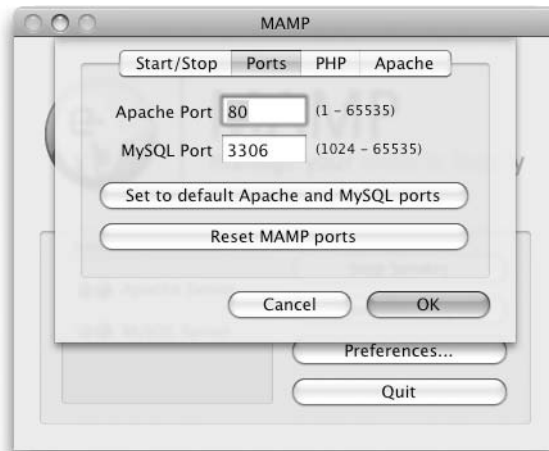


Figure 2-3. The default MAMP settings need to be changed.

4. Click OK. Enter your Mac password when prompted. MAMP should restart both Apache and MySQL with the standard settings for Apache and MySQL.

Creating virtual hosts on Apache

When it's first installed, Apache is capable of hosting only one website, which is identified in a local testing environment by the URL `http://localhost/`. To get around this restriction, it's common practice to develop websites in subfolders of the Apache server root. For example, if you have two sites called `site1` and `site2` and create separate subfolders for them in the server root, you access them in your testing environment as `http://localhost/site1/` and `http://localhost/site2/`. This works perfectly well as long as you use document-relative links all the time. However, if you want to use links relative to the site root, you need to create virtual hosts for each site.

Virtual hosting is a technique that web servers use to host more than one website on the same machine. If you have bought a web-hosting package from a hosting company, it's almost certainly on a shared server that uses virtual hosts. Continuing with the previous example, once you create virtual hosts for `site1` and `site2` in Apache, you can test them locally using `http://site1/` and `http://site2/`. This is essential for testing sites that use links relative to the site root. If you're serious about web development, you should learn sooner or later how to set up virtual hosts in your testing environment. Once you have mastered the technique, it takes only a few minutes to set up each one.

The rest of this section is entirely optional. If you don't want to set up virtual hosts, you can skip it. You can come back and set up virtual hosts at any time.

You can call your virtual hosts whatever you like, as long as you don't use any spaces or characters that would be illegal in a domain name. I always use the same name as the actual website, *without* the top-level domain. For example, for my own site, `http://foundationphp.com/`, I have created a virtual host called `foundationphp` in my local testing setup. This means that I access it as `http://foundationphp/`. It's then a simple matter of clicking in the browser address bar and adding the `.com` to see the live site. Whatever you do, *don't* use the top-level domain as the name of a virtual host in your testing setup. If you do, your computer will always point to the local version of the site and never access the real one on the Internet.

Apache allows you to create as many virtual hosts as you want. It's a two-stage process. First, you tell the operating system the names of the virtual hosts, and then you tell Apache where the files will be located. There are separate instructions for Windows and Mac OS X.

Registering virtual hosts on Windows

Although you can locate your virtual hosts anywhere on your hard drive system, it's a good idea to keep them in a single top-level folder, as this makes it easier to set the correct permissions in Apache. The following instructions assume that all your virtual hosts are kept in a folder called `C:\vhosts` and show you how to create a virtual host called `dwcs4` within that folder.

To edit the necessary files in Vista, you need to select Run as administrator even if you are logged in to an administrator account. For Notepad, go to Start ► All Programs ► Accessories, right-click Notepad, and select Run as administrator from the context menu. Enter your administrator password when prompted. Inside Notepad, select File ► Open and navigate to the relevant file. The Open dialog box in Notepad shows only .txt files, so you need to select All Files (.*) from the drop-down menu at the bottom right of the dialog box.*

1. Create a folder called `C:\vhosts` and a subfolder inside it called `dwcs4`.
2. Open `C:\WINDOWS\system32\drivers\etc\hosts` in Notepad or a script editor and look for the following line at the bottom of the file:

```
127.0.0.1 localhost
```

127.0.0.1 is the IP address that every computer uses to refer to itself.

3. On a separate line, enter `127.0.0.1`, followed by some space and the name of the virtual host. For instance, to set up a virtual host for this book, enter the following:

```
127.0.0.1 dwcs4
```

4. If you want to register any further virtual hosts, add each one on a separate line and point to the same IP address. Save the hosts file and close it.

5. Open the Apache configuration file, httpd.conf. The default location is C:\Program Files\Apache Software Foundation\Apache2.2\conf\httpd.conf. If you installed XAMPP, it should be at C:\xampp\apache\conf\httpd.conf.
6. Scroll down to the Supplemental configuration section at the end of httpd.conf, and locate the following section:

```
462 # Virtual hosts
463 #Include conf/extra/httpd-vhosts.conf
```

7. Apache uses the hash sign (#) to indicate comments in its configuration files. Uncomment the command shown on line 463 in the preceding screenshot by removing the #, like this:

```
Include conf/extra/httpd-vhosts.conf
```

This tells Apache to include the virtual host configuration file, which you must now edit.

8. Save httpd.conf, and close it.
9. Open httpd-vhosts.conf. The default location is C:\Program Files\Apache Software Foundation\Apache2.2\conf\extra\httpd-vhosts.conf. If you installed XAMPP, it should be at C:\xampp\apache\conf\extra\httpd-vhosts.conf. The main part of the file looks like this:

```
13 # You may use the command line option '-S' to verify your virtual host
14 # configuration.
15
16 #
17 # Use name-based virtual hosting.
18 #
19 NameVirtualHost *:80
20
21 #
22 # VirtualHost example:
23 # Almost any Apache directive may go into a VirtualHost container.
24 # The first VirtualHost section is used for all requests that do not
25 # match a ServerName or ServerAlias in any <VirtualHost> block.
26 #
27 <VirtualHost *:80>
28     ServerAdmin webmaster@dummy-host.home
29     DocumentRoot /www/docs/dummy-host.home
30     ServerName dummy-host.home
31     ServerAlias www.dummy-host.home
32     ErrorLog logs/dummy-host.error_log
33     CustomLog logs/dummy-host.access_log common
34 </VirtualHost>
35
36 <VirtualHost *:80>
37     ServerAdmin webmaster@dummy-host2.home
38     DocumentRoot /www/docs/dummy-host2.home
39     ServerName dummy-host2.home
40     ErrorLog logs/dummy-host2.error_log
41     CustomLog logs/dummy-host2.access_log common
42 </VirtualHost>
```

- 10.** Position your cursor in the blank space shown on line 15 in the preceding screenshot, and insert the following four lines of code:

```
<Directory C:/vhosts>
  Order Deny,Allow
  Allow from all
</Directory>
```

This sets the correct permissions for the folder that contains the sites you want to treat as virtual hosts. If you chose a location other than C:\vhosts as the top-level folder, replace the pathname in the first line. Remember to use forward slashes in place of backward slashes. Also surround the pathname in quotes if it contains any spaces.

- 11.** Lines 27–42 in the preceding screenshot are examples of virtual host definitions. They show all the commands that can be used, but only DocumentRoot and ServerName are required. When you enable virtual hosting, Apache disables the main server root, so the first definition needs to reproduce the original server root. You then add each new virtual host within a pair of <VirtualHost> tags, using the location of the site's web files as the value for DocumentRoot and the name of the virtual host for ServerName. If the path contains any spaces, enclose the whole path in quotes. If your server root is located, like mine, at C:\htdocs, and you are adding dwcs4 as a virtual host in C:\vhosts, change the code shown on lines 27–42 so they look like this:

```
<VirtualHost *:80>
  DocumentRoot c:/htdocs
  ServerName localhost
</VirtualHost>
<VirtualHost *:80>
  DocumentRoot c:/vhosts/dwcs4
  ServerName dwcs4
</VirtualHost>
```

For XAMPP, use C:/xampp/htdocs instead of C:/htdocs.

- 12.** Save httpd-vhosts.conf, and restart Apache.

All sites in the server root will continue to be accessible through <http://localhost/sitename/>. Anything in a virtual host will be accessible through a direct address, such as <http://dwcs4/>.

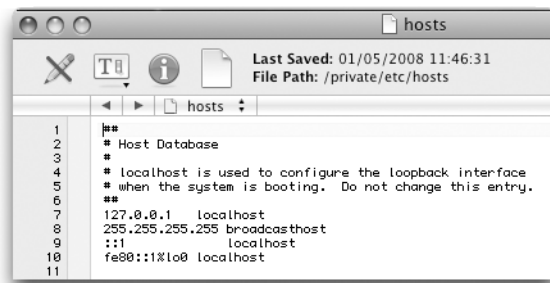
Registering virtual hosts on Mac OS X

The following instructions apply *only* to the preinstalled version of Apache on Mac OS X. To enable virtual hosts with MAMP, I recommend that you invest in MAMP PRO (<http://www.mamp.info/en/mamp-pro/index.html>). It's not free, but it automates the configuration of virtual hosts and other aspects of your development environment.

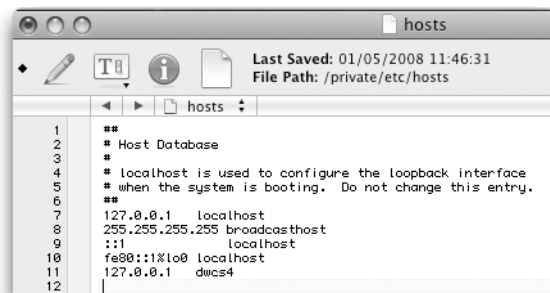
You need to edit hidden files. The simplest way to do this is to use a specialized script editor. I recommend using either BBEdit (<http://www.barebones.com>) or TextWrangler (a free, cut-down version of BBEdit available from the same location).

The following instructions assume that all your virtual hosts are kept in the Sites folder in your Mac home folder, and show how to create a virtual host called `dwcs4` within that folder. Setting up virtual hosts on a Mac changed substantially between OS X 10.4 and 10.5, so there are separate instructions for each version. First, Mac OS X 10.5:

1. Open BBEdit or TextWrangler and select File ► Open Hidden. In the Open dialog box, select All Files from the Enable drop-down menu. Then navigate to Macintosh HD:private:etc:hosts and click Open.
2. This opens a system file, so you need to unlock it by clicking the icon of a pencil with a line through it on the left side of the toolbar, as shown in the following screenshot:



3. You will be told that the document is owned by “root” and asked to confirm that you want to unlock it. Click Unlock. This removes the line through the pencil and readies the file for editing.
4. Place your cursor on a new line at the end of the file, and type `127.0.0.1`, followed by a space and the name of the virtual host you want to create. To create a virtual host for this book called `dwcs4`, it should look like this:



5. Save the file. Because it’s owned by root, you will be prompted to enter your Mac password. You now need to tell Apache about the virtual host.
6. Use BBEdit or TextWrangler to open the main Apache configuration file, `httpd.conf`. It’s a system file, so you need to open and unlock it in the same way as the `hosts` file. It’s located at Macintosh HD:private:etc:apache2:httpd.conf.

7. Scroll down to around line 460 and locate the following lines:

```
# Virtual hosts
#Include /private/etc/apache2/extra/httpd-vhosts.conf
```

8. Remove the hash sign (#) from the beginning of the second of these two lines so it looks like this:

```
Include /private/etc/apache2/extra/httpd-vhosts.conf
```

This enables the configuration file for virtual hosts, which now needs to be edited.

9. Use BBEdit or TextWrangler to open `httpd-vhosts.conf`. Again, it's a system file, so it needs to be handled the same way as the previous two files. The file is located at `Macintosh HD:private:etc:apache2:extra:httpd-vhosts.conf`.
10. The section of the file that you're interested in is shown in the following screenshot:



```
17 # Use name-based virtual hosting.
18 #
19 NameVirtualHost *:80
20
21 #
22 # VirtualHost example:
23 # Almost any Apache directive may go into a VirtualHost container.
24 # The first VirtualHost section is used for all requests that do not
25 # match a ServerName or ServerAlias in any <VirtualHost> block.
26 #
27 <VirtualHost *:80>
28     ServerAdmin webmaster@dummy-host.example.com
29     DocumentRoot "/usr/docs/dummy-host.example.com"
30     ServerName dummy-host.example.com
31     ServerAlias www.dummy-host.example.com
32     ErrorLog "/private/var/log/apache2/dummy-host.example.com-error_log"
33     CustomLog "/private/var/log/apache2/dummy-host.example.com-access_log" common
34 </VirtualHost>
35
36 <VirtualHost *:80>
37     ServerAdmin webmaster@dummy-host2.example.com
38     DocumentRoot "/usr/docs/dummy-host2.example.com"
39     ServerName dummy-host2.example.com
40     ErrorLog "/private/var/log/apache2/dummy-host2.example.com-error_log"
41     CustomLog "/private/var/log/apache2/dummy-host2.example.com-access_log" common
42 </VirtualHost>
```

Lines 27–42 are examples of virtual host definitions. You need to replace these with your own definitions. When you enable virtual hosting, Apache disables the main server root, so the first definition needs to reproduce it.

You don't need all the options shown in the examples, so replace the code shown on lines 27–42 of the preceding screenshot with the following:

```
<VirtualHost *:80>
    DocumentRoot "/Library/WebServer/Documents"
    ServerName localhost
</VirtualHost>
<VirtualHost *:80>
    DocumentRoot "/Users/username/Sites/dwcs4"
    ServerName dwcs4
</VirtualHost>
```

Replace `username` in the second definition with your own Mac username.

11. Save all the files you have edited, and restart Apache by going to Sharing in System Preferences ► Internet & Network, deselecting Web Sharing, and selecting it again. You should now be able to access the virtual host with the URL `http://dwcs4/`.

Follow these instructions for Mac OS X 10.4:

1. Open NetInfo Manager, which is in the Utilities subfolder of Applications.
2. Click the lock at the bottom left of the dialog box that opens, and enter your administrator's password when prompted.
3. Select machines, then localhost, and click the Duplicate icon. When prompted, confirm that you want to make a copy.
4. Highlight the copy, and double-click the name in the lower pane, as shown in the following screenshot.



5. Change localhost copy to whatever you want to call the virtual host. For example, to create a virtual host for this book, enter dwcs4.
6. Click any of the other entries in the left column of the top pane. The operating system will ask you twice if you really want to make the changes. You do. This registers the name of the virtual host with your computer.
7. Repeat steps 3–6 for any other virtual hosts you want to create. When you have finished, click the lock icon in the bottom-left corner of the NetInfo Manager, and close it.

8. Open BBEdit or TextWrangler, and select File ► Open Hidden. In the Open dialog box, select All Files from the Enable drop-down menu, and open Macintosh HD:etc:httpd:httpd.conf.
9. Scroll almost to the bottom of httpd.conf, and locate the following section:

```

1073 #
1074 # Use name-based virtual hosting.
1075 #
1076 #NameVirtualHost *:80
1077
1078 #
1079 # VirtualHost example:
1080 # Almost any Apache directive may go into a VirtualHost container.
1081 # The first VirtualHost section is used for requests without a known
1082 # server name.
1083 #
1084 #<VirtualHost *:80>
1085 #     ServerAdmin webmaster@dummy-host.example.com
1086 #     DocumentRoot /www/docs/dummy-host.example.com
1087 #     ServerName dummy-host.example.com
1088 #     ErrorLog logs/dummy-host.example.com-error_log
1089 #     CustomLog logs/dummy-host.example.com-access_log common
1090 #</VirtualHost>

```

10. Click the pencil icon at the top left of the editor window, and confirm that you want to unlock the document, entering your administrator password when prompted. Uncomment the command shown on line 1076 in the screenshot by removing the hash sign (#). This enables virtual hosting but disables the main server root, so the first virtual host needs to reproduce the Mac's server root. The example (on lines 1084–1090) is there to show you how to define a virtual host. The only required commands are DocumentRoot and ServerName. After uncommenting the NameVirtualHost command, your first definition should look like this:

```

NameVirtualHost *:80
<VirtualHost *:80>
    DocumentRoot /Library/WebServer/Documents
    ServerName localhost
</VirtualHost>

```

11. Add any further definitions for virtual hosts. To create one for this book, use this (replace *username* with your own Mac username):

```

<VirtualHost *:80>
    DocumentRoot /Users/username/Sites/dwcs4
    ServerName dwcs4
</VirtualHost>

```

12. Save httpd.conf, and restart Apache. All sites in Macintosh HD:Library:WebServer:Documents can still be accessed using http://localhost/ and those in your Sites folder using http://localhost/~username/sitename/, but named virtual hosts can be accessed directly, such as http://dwcs4/. Of course, a site must exist in the location you defined before you can actually use a virtual host.

Registering virtual directories on IIS

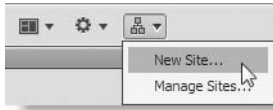
Windows Vista uses IIS 7, which lets you set up separate websites, each with its own server root, just like Apache virtual hosts. However, the version of IIS that runs in Windows XP does not support virtual hosts. Instead, you can set up virtual directories, but localhost

always remains the basic address of the web server, so you cannot use root-relative links. The main advantage of using virtual directories is that they avoid the need to locate all web files in the default IIS server root at C:\Inetput\wwwroot.

To set up a virtual directory in IIS 6 on Windows XP, open the Internet Information Services panel (Start ► Control Panel ► Administrative Tools ► Internet Information Services), highlight Default Web Server, right-click, and select New ► Virtual Directory. A wizard will walk you through the process. If you create a virtual directory called dwcs4, the URL becomes http://localhost/dwcs4/.

Creating the site definition

By this stage, you should have decided where you are going to store your local files. The setup process in Dreamweaver is basically the same whether you test your PHP files locally or on your remote server.



There are several ways to open the Site Definition dialog box. If the Dreamweaver Welcome screen is open, you can choose Dreamweaver Site from the bottom of the Create New column. However, it's probably more convenient to choose New Site from the Site menu or from the Site icon on the Application bar (see alongside). Another convenient way is to select Manage Sites from the bottom of the site list at the top left of the Files panel.

Dreamweaver has been designed with both beginners and more advanced users in mind, so you may see either the basic dialog box shown on the left of Figure 2-4 or the advanced one on the right.

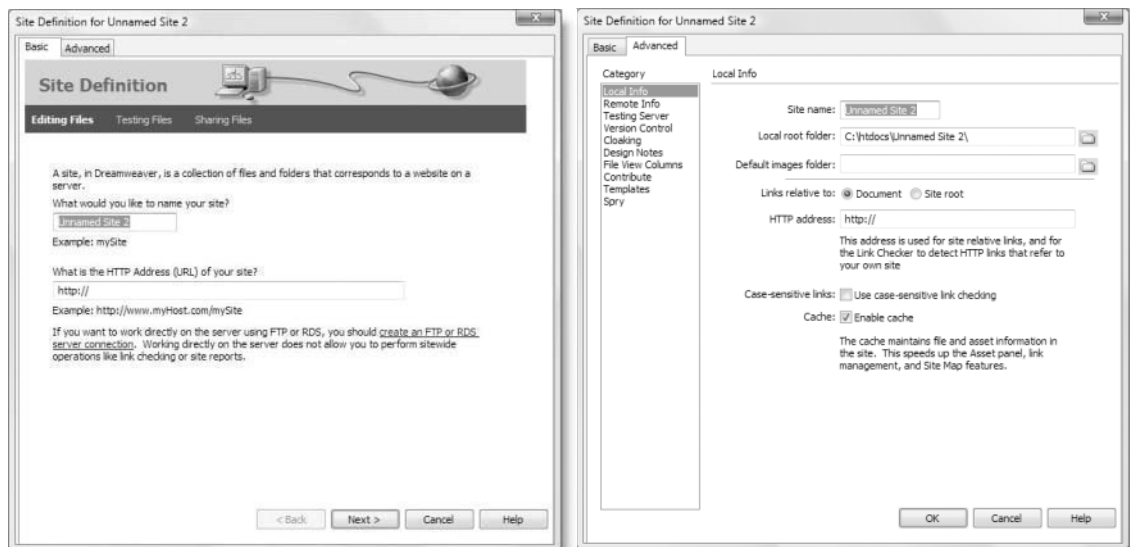


Figure 2-4. The Site Definition dialog box has two interfaces: Basic (left) and Advanced (right).

The Basic dialog box sets up only the bare essentials, so it's better to use the Advanced one. If you see the screen on the left of Figure 2-4, click the Advanced tab at the top left (it's in the center of the Mac version).

If you select Manage Sites from the Files panel, you will be presented with the dialog box shown in Figure 2-5. This lists the sites that you have already defined in Dreamweaver. The buttons on the right let you perform a variety of management functions, as described in the “Managing Dreamweaver sites” section later in the chapter. To create a new site, click the New button at the top right and select Site from the menu that appears.

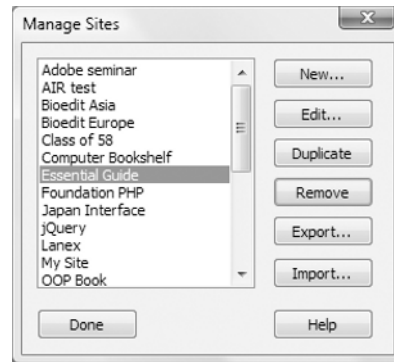


Figure 2-5.
The Manage Sites dialog box lets you create a new site or edit an existing one.

Telling Dreamweaver where to find local files

The first stage of site definition involves defining the basic details of the site. Open the Site Definition dialog box, and make sure the Advanced tab is selected. If necessary, select Local Info from the Category column on the left. You should see the same screen as shown on the right side of Figure 2-4.

Let's take a look at what each option means, with particular reference to defining a PHP site for use with this book.

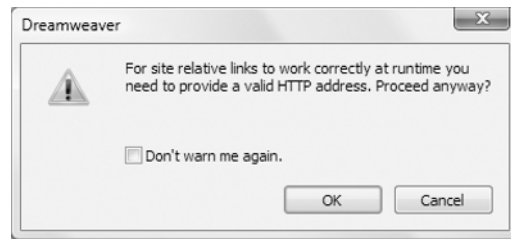
- **Site name:** This identifies the site within Dreamweaver. The name appears in the drop-down menu at the top of the Files panel and in the Manage Sites dialog box (Figure 2-5), so it needs to be reasonably short. It's used only within Dreamweaver, so spaces are OK. I used Dreamweaver CS4.
- **Local root folder:** This is the top-level folder of the site. Everything should be stored in this folder in exactly the same hierarchy as you want to appear on the live website. For a static site using .html pages only or when using a remote server to test PHP, this folder can be anywhere on your computer. When testing a PHP site locally, this folder should be inside your server root (see the “Finding the testing server root” section earlier in this chapter), a virtual host, or a virtual directory (IIS only). Click the folder icon to the right of the Local root folder field and navigate to

the appropriate location on your hard disk. If the folder doesn't exist, navigate to the parent folder, and then click Create New Folder in the Choose local root folder dialog box.

With large sites, it's sometimes convenient to create a site definition in Dreamweaver for just part of the site. If the local root folder is already in another defined site, Dreamweaver warns you that some functions, such as site synchronization, won't work. However, it won't prevent you from creating the subsite.

2

- **Default images folder:** This field is optional, but is very useful if you plan to use images that are on other parts of your file system or even in other Dreamweaver sites. Whenever you insert an image in a web page, Dreamweaver automatically copies it to this folder and creates the correct link in the `` tag's `src` attribute. To set this option, click the folder icon to the right of the Default images folder field, navigate to the local root folder that you selected for the previous option, and select the `images` folder. If the folder doesn't exist, click the Create New Folder button to create it.
- **Links relative to:** This option lets you select the default style of links used in the site (see the "Choosing document- or root-relative links" section earlier in the chapter). Unless your testing server and remote server both run on Apache, I strongly advise you to accept the default Document.
- **HTTP address:** This field should contain the URL of the final site on the Internet. If you are using the site only for local testing, you can leave this field empty. If you have selected root-relative links, Dreamweaver will display the following warning:



You can safely ignore this warning for local testing, and click OK. However, it is important to get the URL correct for remote testing or a site that you plan to deploy on the Internet.

- **Case-sensitive links:** The vast majority of PHP websites are hosted on Linux servers, which treat `products.php` and `Products.php` as completely different file names. If you select this option, Dreamweaver checks that internal links match the case of

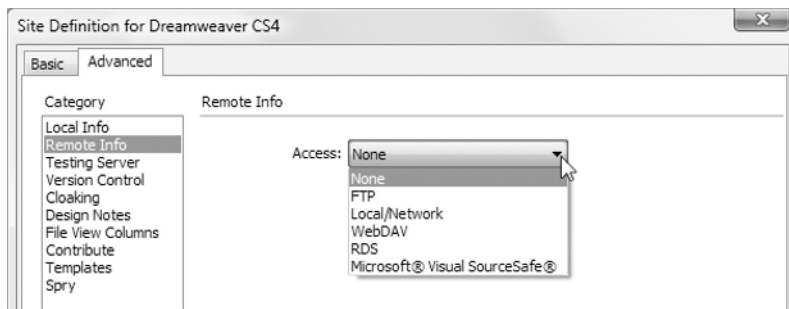
file names when you run Site ► Check Links Sitewide. I recommend selecting this option to maintain the internal integrity of your site.

- **Cache:** As the Site Definition dialog box explains, this speeds up various aspects of site management in Dreamweaver. Very large sites (with several hundred pages) tend to slow down dramatically if the site cache is enabled. However, with a PHP site, you should draw content from a database into a dynamically generated page, rather than create a new page every time. I suggest that you leave this option selected, and disable it only if you run into performance problems.

The Local Info category is the only one you need to complete in order to get to work with static pages (in other words, ones that don't use a server-side technology) on your local computer. If you're not ready to upload files to your live website or work with PHP, just click OK in the Site Definition dialog box, and then click Done to close the Manage Sites dialog box. Otherwise, continue with the next sections.

Telling Dreamweaver how to access your remote server

When you first open the Remote Info category in the Site Definition dialog box, you're presented with a single drop-down menu labeled Access. It has six options, as shown in the following screenshot (the final option—Microsoft Visual SourceSafe—is not available in the Mac version).



Choosing an access option

First, let's take a look at the Access options:

- **None:** Choose this if you don't plan to deploy the site on the Internet, or if you don't want to set up your remote server immediately. If you choose this option, you can skip ahead to the "Defining the testing server" section.
- **FTP:** This is the most common choice. It sets up Dreamweaver's built-in File Transfer Protocol (FTP) program to communicate with your remote server.
- **Local/Network:** This allows you to deploy your live website to another folder on your local computer or network. This is normally done only by organizations that run their own live web servers.

- WebDAV: This uses the Web-based Distributed Authoring and Versioning (WebDAV) protocol to communicate with the remote server. It requires a remote server that supports the WebDAV protocol.
- RDS: This uses Remote Development Services (RDS), which is supported only by ColdFusion servers. You cannot use it with a PHP site unless the server also supports ColdFusion.
- Microsoft Visual SourceSafe: This requires access to a Microsoft Visual SourceSafe database. It is not appropriate for the Dreamweaver PHP MySQL server model.

Since FTP is the most common method of connecting to a remote server, that's the only one I'll describe. Click the Help button at the bottom of the Remote Info category of the Site Definition dialog box for detailed descriptions of the options for the other methods.

Using FTP

When you select the FTP option from the Access drop-down menu, the Remote Info category of the Site Definition dialog box presents you with the options shown in Figure 2-6. Most of them are straightforward, but I'll describe each one briefly.

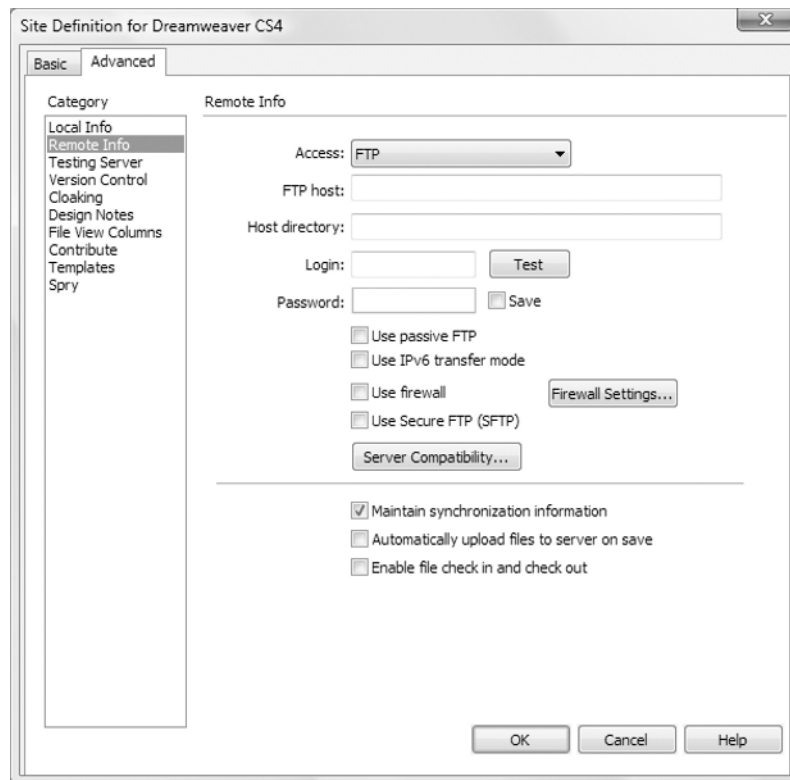


Figure 2-6. The FTP options for the Remote Info category of the Site Definition dialog box

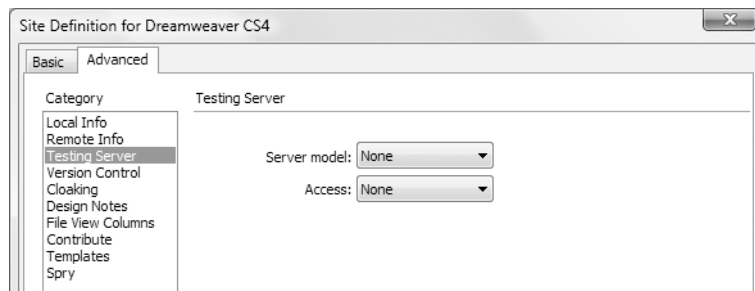
- **FTP host:** Enter your remote server's FTP address in this field. You should normally get this from your hosting company. It usually takes either of the following forms: `ftp.example.com` or `www.example.com`.
- **Host directory:** This is the pathname of the top level of your website. The important thing to realize is that the directory (folder) that you enter in this field should contain only those files that will be accessible to the public through your site's URL. Often it will be named `htdocs`, `public_html`, or `www`. If in doubt, ask your hosting company or server administrator.
- **Login:** This is the username given to you by your hosting company or server administrator.
- **Password:** Enter your remote server password in this field. Dreamweaver displays your password as a series of dots. It also automatically saves your password, so deselect the `Save` checkbox if you want to be prompted for the password each time you connect to the remote server. Click the `Test` button to make sure that Dreamweaver can connect successfully. If the test fails, make sure `Caps Lock` isn't turned on, as passwords are normally case-sensitive. Other reasons for failure include being behind a firewall, so check the remaining options before trying again. Many antivirus programs include a software firewall, as does Windows Vista, so you should also check whether it's preventing Dreamweaver from accessing the Internet.
- **Use passive FTP:** Try this option if a software firewall prevents you from connecting to the remote server. For more details, see <http://www.adobe.com/go/15220>.
- **Use IPv6 transfer mode:** This option is designed to prepare Dreamweaver for the future. Select this option only if you have been told that your remote FTP server uses Internet Protocol version 6 (IPv6).
- **Use firewall:** You can normally ignore this option unless you are behind a corporate firewall. The `Firewall Settings` button opens the `Site Preferences` dialog box. Enter the firewall host and firewall port (if it's different from 21) in the appropriate fields, and click `OK` to return to the `Site Definition` dialog box. If you are using a software firewall, such as Norton Internet Security or ZoneAlarm, you need to set permission for Dreamweaver to access the Internet in the software firewall's configuration settings rather than here.
- **Use Secure FTP (SFTP):** Secure FTP (SFTP) gives you a more secure connection, but it is not supported by all servers. Selecting this option automatically disables these other options: `Use passive FTP`, `Use IPv6 transfer mode`, `Use firewall`, `Firewall Settings`, and `Server Compatibility`.
- **Server Compatibility:** Click this button if you are still having problems connecting through FTP. The two options in the dialog box that opens are self-explanatory.
- **Maintain synchronization information:** This is selected by default and enables you to synchronize your remote and local files through the `Files` panel. However, it's not very reliable, particularly if you live in a part of the world that observes daylight saving time.

- Automatically upload files to server on save: Do *not* select this option. You should always test files locally before uploading them to your remote server. Otherwise, all your mistakes will go public. It overwrites your original files, so you can no longer use them as backup.
- Enable file check in and check out: Select this option only if you are working in a team and want to use Dreamweaver's Check In/Check Out system. For more information, launch Dreamweaver Help (press F1) and select Check In/Check Out from the Index, or go to <http://www.adobe.com/go/15447>. All team members must have this option enabled and must always use Dreamweaver to edit files. Failure to do so results in chaos. This option should be used with extreme caution.

After you have completed the Remote Info category, select Testing Server from the Category list on the left of the Site Definition dialog box.

Defining the testing server

When you first open the Testing Server category of the Site Definition dialog box, it looks similar to the Remote Info category in its initial state, but with two drop-down menus instead of one, as shown in the following screenshot.



This is probably the most important dialog box when building dynamic sites in Dreamweaver. It's quite easy to fill in, but if you get the details wrong, Dreamweaver cannot communicate with any of your databases.

Activate the Server model drop-down menu, and select PHP MySQL. What you choose for Access depends on whether you want to test your PHP pages locally or by using your remote server. The options are different, so I'll cover them separately.

Selecting options for local testing

The Access drop-down menu determines how you communicate with the testing server. If you have a local test environment on your computer or another computer on a local area network (LAN), choose Local/Network. This reveals two options that Dreamweaver attempts to fill in automatically. Figure 2-7 shows what happened when I had defined the local root folder in the Local Info category as a virtual host on Windows.

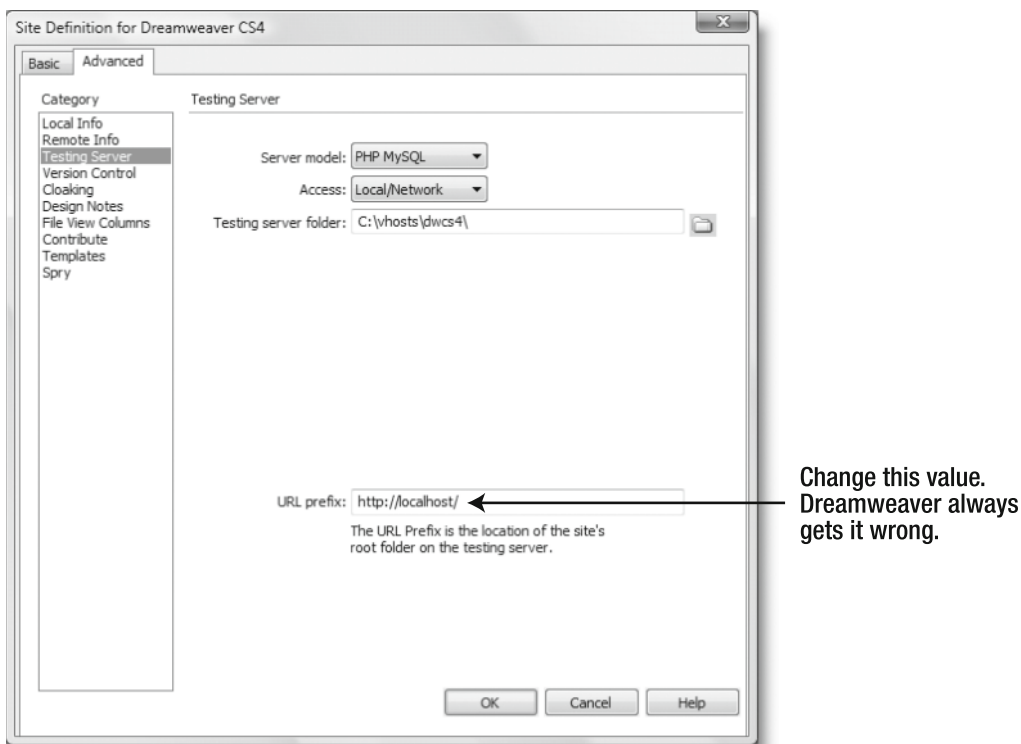


Figure 2-7. Dreamweaver attempts to fill in the testing server details automatically.

Dreamweaver usually gets the value for Testing server folder correct, but invariably gets URL prefix wrong. Getting both correct is crucial, so let's take a look at what they represent.

Testing server folder and URL prefix must both point to the same location. The value you enter in Testing server folder is the physical path to your site root (where you keep the home page). The URL prefix is the address you would enter in a browser address bar to get to the same page (minus the page name).

The value for Testing server folder should normally be the same folder that you selected as the Local root folder in the Local Info category. The only exception is if you want to use a testing server elsewhere on your local network. In this case, click the folder icon to the right of the field to browse to the correct location.

The value for URL prefix depends on how you have set up your testing environment. If your testing server folder is in the server root or a virtual directory, it will be `http://localhost/sitename/`. If you are using a virtual host, it will simply be `http://sitename/`. If the testing server is on another computer on a local network, replace `localhost` with the correct IP address.

It's critical that URL prefix is set correctly, as it controls all dynamic aspects of Dreamweaver. Because so many people seem to get this wrong, Table 2-1 shows the values for Testing server folder and URL prefix for the various scenarios described earlier.

Table 2-1. Testing server folder and URL prefix values for various scenarios

Scenario	Testing server folder	URL prefix
Site in a subfolder of the Apache server root of the same machine on Windows	C:\htdocs\dwcs4\ or C:\xampp\htdocs\dwcs4\	http://localhost/dwcs4/
Site in a virtual host called dwcs4 on Windows	C:\vhosts\dwcs4\	http://dwcs4/
Site in an IIS virtual directory on Windows	Can be anywhere	http://localhost/dwcs4/
Site in a subfolder of the main server root of the same machine on a Mac	Macintosh HD:Library: WebServer:Documents:dwcs4	http://localhost/dwcs4/
Site in a subfolder of your Sites folder of the same machine on a Mac	Macintosh HD:Users: username:Sites:dwcs4	http://localhost/~username/ dwcs4/
Site in a virtual host called dwcs4 on a Mac	Macintosh HD:Users:username: Sites:dwcs4	http://dwcs4/
Using MAMP on a Mac with the default Apache and MySQL ports	Macintosh HD:Users:username: Applications:MAMP:htdocs:dwcs4:	http://localhost/dwcs4/
Using MAMP on a Mac with the MAMP default ports	Macintosh HD:Users:username: Applications:MAMP:htdocs:dwcs4:	http://localhost:8888/dwcs4/

Selecting options for remote testing

I strongly advise against using a remote server for testing PHP pages. In addition to the advantages of a local testing server mentioned at the beginning of this chapter, you should also take into consideration the fact that using a remote server for testing overwrites existing files. You can get around this problem by using temporary files for previewing (see the “Setting options for Preview in Browser” section at the end of this chapter), but you can’t use temporary files to test links or work with a database. Of course, you may still decide to use your remote server as the Dreamweaver testing server, and this section describes the necessary settings.

The Access drop-down menu in the Testing Server category offers fewer options than the Remote Info category, because RDS and Microsoft SourceSafe are not appropriate for working with the Dreamweaver PHP MySQL server model. The most common choice is FTP. Dreamweaver is intelligent enough to copy across the main details from the Remote Info category, and it presents you with the dialog box shown in Figure 2-8. Although most details should be correct, the URL prefix is almost certain to need editing.

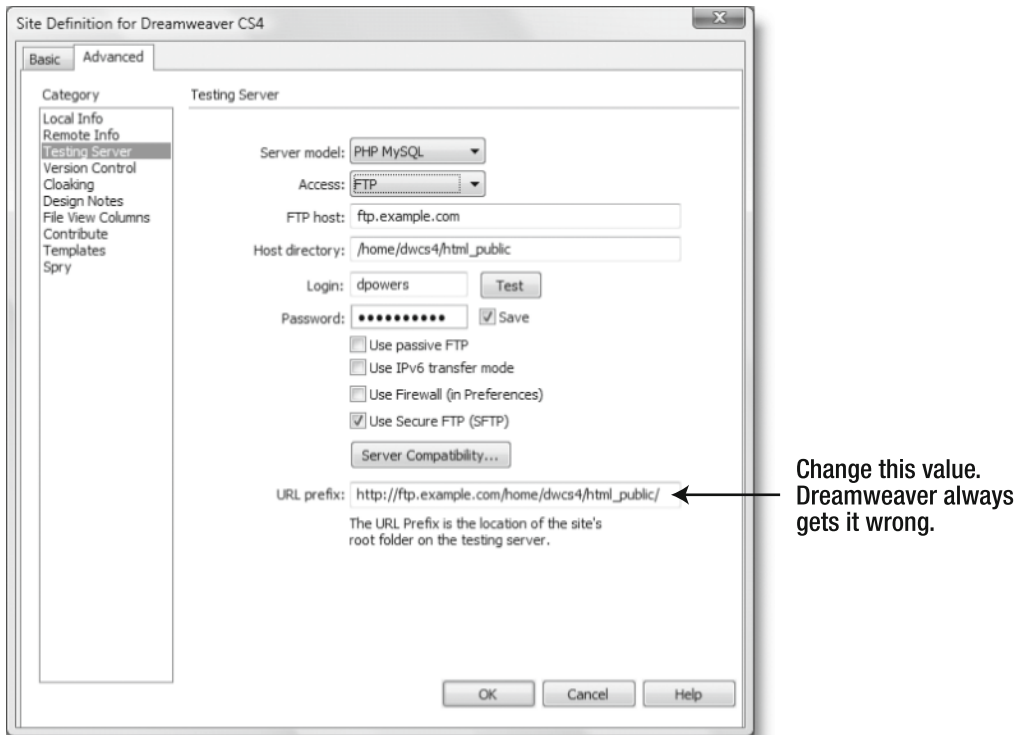


Figure 2-8. When you select a remote server for testing, Dreamweaver copies details from the Remote Info category, but you normally need to change at least the URL prefix.

As you can see from Figure 2-8, Dreamweaver incorrectly combines the values in the FTP host and Host directory fields. This won't work, and the URL prefix value must be changed.

It's vital that the URL prefix and Host directory fields point to the same place. However, this does *not* mean that the values should be the same. The distinction is as follows:

- Host directory: This is the pathname that the FTP program uses for the top level of your site.
- URL prefix: This is the address that anyone surfing the Internet uses to reach the top level of your site. In other words, it's normally `http://` followed by the domain name and a trailing slash.

So, if `/home/dwcs4/html_public/index.php` is your home page, and users access it by typing `http://www.example.com/index.php` in their browser address bar, the correct value for URL prefix should look like this:

`http://www.example.com/`

In Figure 2-8, notice that even though the Use Secure FTP (SFTP) checkbox is selected, the three checkboxes above and the Server Compatibility button are not grayed out, as in the Remote Info category. This is a known bug in Dreamweaver. Make sure you don't accidentally select them if you're using SFTP. The settings should be the same as in the Remote Info category.

Setting other site options

The basic site definition is now complete. To save the site definition, click OK in the Site Definition dialog box, and then click Done to close the Manage Sites dialog box. However, there are seven more categories in the Site Definition dialog box. Most of the time, you can leave these at their default values.

- **Version control:** Dreamweaver CS4 now offers integration with Subversion (<http://subversion.tigris.org/>), one of the most popular open source version control systems. See the next section for details on using this option for your site.
- **Cloaking:** Some developers like to keep source files, such as .fla files for Flash movies or .psd files for Photoshop, in the same folder as their site. To prevent them from being uploaded to your remote server when uploading or synchronizing a complete site, you can use Dreamweaver's cloaking feature. The Cloaking category of the Site Definition dialog box lets you automatically cloak all files with specific file name extensions. There are just two options. The first one, Enable cloaking, is selected by default. To cloak specific types of files automatically, select the Cloak files ending with checkbox and enter the file name extensions as a space-delimited list. In new sites created in Dreamweaver CS4, the field is prepopulated with .fla .psd (in sites migrated from older versions of Dreamweaver, it's prepopulated with .png .fla).

To cloak individual files or folders, select them in the Files panel, right-click, and select Cloaking ► Cloak from the context menu. The ability to cloak individual files is new to Dreamweaver CS4.

- **Design Notes:** Design notes serve a number of different purposes. One is to store notes about individual files. This is mainly of use in a team environment, where different members can add notes regarding the status of the file (draft, first revision, things remaining to be done, and so on). Dreamweaver also creates design notes automatically to store information about related files, file synchronization, and locally created variables. For instance, if you create an image in Fireworks and import it into Dreamweaver, the location of the original .png file is stored in a design note, enabling you to open it directly from the Document window if you want to edit the original image. By default, Dreamweaver enables design notes and creates them in a hidden folder called _notes inside most folders within your site. If you don't want design notes, you can turn them off in the Design Notes category of the Site Definition dialog box. If you're working in a team environment, there's an option to upload design notes to the remote server.
- **File View Columns:** This lets you customize the look of the Files panel.

- **Contribute:** This allows you to use rollback and event logging when developing the site to be updated with Contribute (<http://www.adobe.com/products/contribute/>).
- **Templates:** This is for backward compatibility with older versions of Dreamweaver templates. It should be left at its default setting (enabled).
- **Spry:** Adobe's Ajax framework, Spry, relies on code libraries that need to be uploaded to your remote server. By default, Dreamweaver inserts these files in a folder called `SpryAssets` at the top level of your site root. For most people, this is ideal. However, if you want to locate the code libraries elsewhere, specify the folder name in the `Spry` category of the `Site Definition` dialog box. This allows Dreamweaver to update or remove the files when you make changes to elements that use Spry.

Using version control with Subversion

Some form of **version control** is standard in team environments, but it's something that individual developers either don't know about or tend to treat like regular backups of hard disks—you know you ought to do it, but never quite get around to it. If you have ever made changes to a file and wished you could roll back to the original, you need version control. In one respect, version control acts as a database, storing project files at different stages of development. Instead of overwriting the original file each time, it stores a snapshot of each stage. It also allows different people to work simultaneously on separate versions of the same document, review each other's changes, and merge them.

Although this is a typical team development scenario, it can also be useful for individual developers. Say you normally work on a desktop computer, but occasionally use a laptop when you're on the move. By storing your files in a repository, you can always have access to the most current version, regardless of which computer you're using. You can also keep different versions of projects. And if you commit files to the repository on a regular basis, you can experiment with a file and roll back to a previous version if you don't like the changes. Once you get into the habit of using version control, you'll wonder how you ever did without it.

As mentioned in the previous section, Dreamweaver CS4 lets you integrate with Subversion. Note that Dreamweaver is not a full Subversion client. It offers a limited range of version control functions. Nevertheless, these functions are extremely useful for keeping track of changes in a site's files, whether you're working on your own or in a team.

At the time of this writing, Dreamweaver CS4 supports only Subversion 1.4. For details, see the Adobe TechNote at <http://www.adobe.com/go/kb406661>.

Subversion is preinstalled on Mac OS X 10.5, and Windows users can install a very user-friendly tool called TortoiseSVN (<http://tortoisesvn.tigris.org/>), which makes it easy to set up a Subversion repository. If you don't want to go to the trouble of configuring everything yourself, there are hosted Subversion repositories (some of them free); for example, see <http://beanstalkapp.com/> and <http://cvsdude.com/product.pl>. You can also download a free book called *Version Control with Subversion* from <http://svnbook.red-bean.com/>.

Registering a site with a Subversion repository

Once you have set up a Subversion repository, adding a Dreamweaver website to it is easy. The following instructions describe the process (refer to Figure 2-9 to see the settings in the dialog box):

1. In the Advanced tab of the Site Definition dialog box, select Version Control from the Category list on the left.
2. In the Access field, select Subversion from the drop-down menu (there are only two options: None and Subversion).
3. In the Protocol field, select the method of connection to the repository from the drop-down menu. There are four options: HTTP, HTTPS, SVN, and SVN+SSH. The choice depends on how the repository has been set up.
4. Enter the domain name where the repository resides in the Server address field. If it's on the same computer, enter localhost. Do *not* prefix this with http://.
5. In the Repository path field, enter the path to the project in which you want to store the site. The project doesn't need to exist in the repository. As you can see in Figure 2-9, I have put a forward slash at the beginning of the path. This is optional; Dreamweaver accepts the path with or without a leading slash.

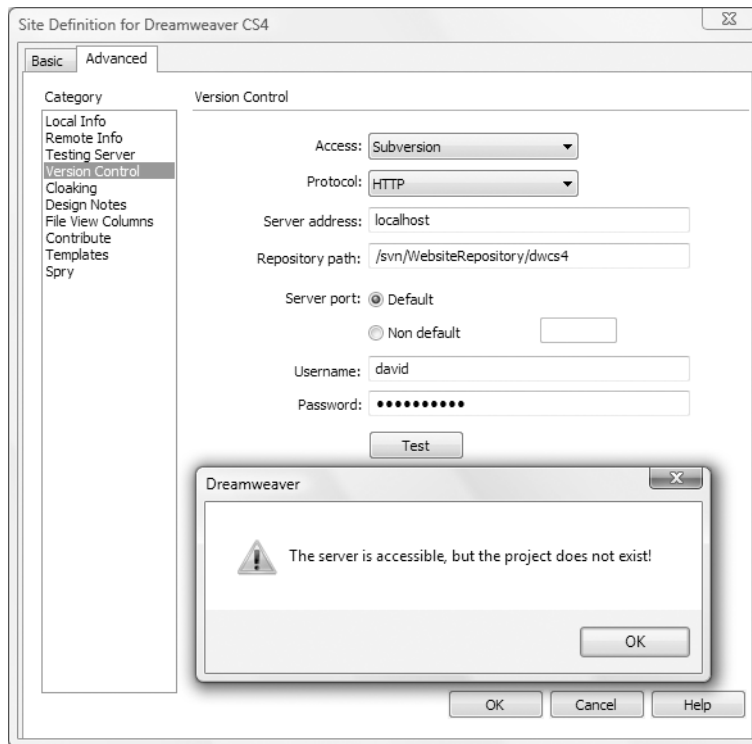
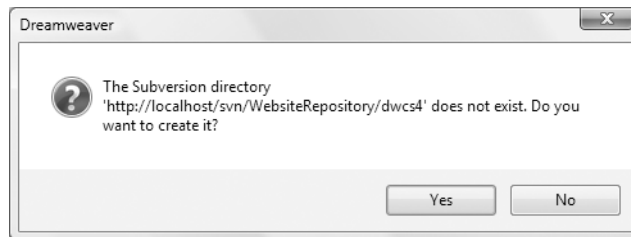


Figure 2-9. As long as Dreamweaver can access the server, you can add a new project to the repository.

6. If the Subversion repository uses a nonstandard port, select the Non default radio button and enter the port number in the field alongside. Otherwise, leave Server port set to Default. This uses the standard Subversion port (3690).
7. Enter the repository username and password in the appropriate fields. If the repository doesn't have user accounts, leave both fields blank.
8. Click Test to make sure Dreamweaver can access the repository. If Dreamweaver connects successfully and the project already exists, you should see an alert with the following message: Server and project are accessible! If the project hasn't yet been created, you should see the message shown in Figure 2-9.
9. Click OK to dismiss the alert.
10. If you need to make changes to any other categories of the Site Definition dialog box, do so, and then click OK to save the changes, and then click Done to dismiss the Manage Sites dialog box.
11. If you entered the name of a new project in step 5, you should see the following alert:



If you click Yes, Dreamweaver should connect to the repository, create the new project, and install the necessary Subversion files in your site.

Subversion treats your site as the working version and keeps track of your files by creating .svn folders in every folder of the site. These folders are normally hidden in the Dreamweaver Files panel, but you can see them in Windows Explorer or Finder on a Mac. Do not delete them unless you no longer want to keep track of the project in Subversion.

If the folder where you defined the site already contains files and folders, Dreamweaver adds a plus (+) icon to the left of each name in the Files panel, as shown in Figure 2-10.

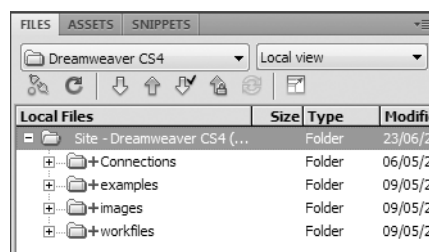


Figure 2-10. Items that haven't yet been added to the repository are marked with a plus icon in the Files panel.

If you're creating a new site in an empty folder, the same icon will be added to each new file or folder that you create. This indicates that it hasn't yet been added to the repository. The next section describes how to commit new files and changes to the repository.

If the site already exists in the Subversion repository, select the Site folder at the top of the Files panel, right-click, and select Version Control ► Get Latest Versions from the context menu. Dreamweaver connects to the repository and downloads the most up-to-date version of each file as your working copy.

Committing new files and changes to the repository

New files and folders that haven't yet been added to the repository are marked with a plus icon in the Files panel. Files in your working copy that have been edited since they were retrieved from the repository are indicated by a check mark alongside the file name, as shown in Figure 2-11. The following instructions explain how to commit new files and changes to the repository.

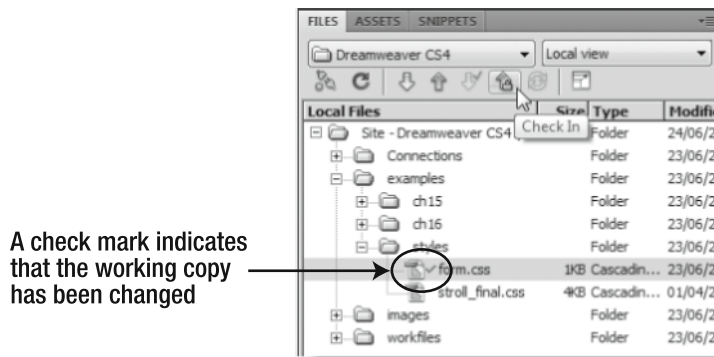


Figure 2-11. The Files panel marks files that have been edited since being checked out of the repository.

1. When you're ready to add new or edited files to the repository, select them in the Files panel. You can select multiple files with Shift-click or Ctrl/Cmd-click. A quick way to select the entire site is to click the Site folder at the top of the panel, as shown in Figure 2-10. With the files selected, click the Check In icon at the top of the Files panel, as shown in Figure 2-11. The icon is an up arrow with a gold padlock.

Don't confuse the Check In icon with the Put File(s) icon two icons further left. Both look very similar. The up arrow without a padlock is used for uploading files to your remote server.

2. The CheckIn dialog box opens, as shown in Figure 2-12. This displays a list of files that will be added to the repository. You can exclude any files by selecting them in the top pane and clicking the red circle with the horizontal slash. This changes the

Action to Ignore. If you change your mind, you can restore the item to the list by clicking the up arrow alongside. Before committing the files to the Subversion repository, you should enter a brief description of the changes in the Commit Message field. This makes it easier to identify stored versions later.

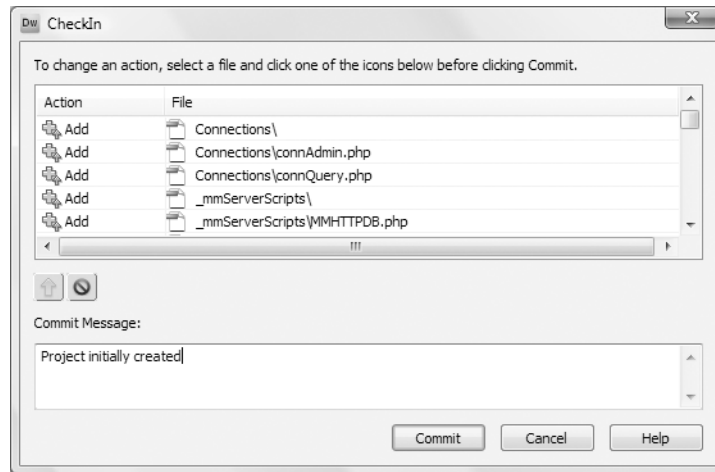
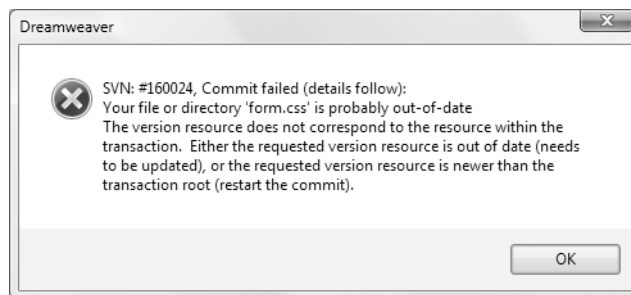


Figure 2-12. When committing files to the repository, it's a good idea to add a message summarizing the changes.

3. Click **Commit** to upload the files to the Subversion repository. If there are no problems, the icons alongside the file and folder names in the Files panel disappear, indicating that all local files have been committed to the repository. If any changes in a working copy conflict with the latest copy stored in the repository, Dreamweaver displays the following alert:



You need to resolve any conflicts, as described in the next section, before you can commit the edited file to the repository.

Viewing revisions and resolving conflicts

Subversion doesn't simply upload your new version of a file to the repository. It compares it with the latest version held in the repository and merges the changes. This is vital when more than one person is working on the same file. If two people check out a file and edit it, you don't want any changes made by one of them to be overwritten by the other. If there's no conflict, both sets of edits are merged into the latest version. However, if both people make incompatible changes (for example, one changes the color in a style rule to red and the other changes it to green), you need to view the revisions and resolve the conflict manually.

To view the revisions made to a file, select it in the Files panel, right-click, and select Version Control ► Show Revisions from the context menu. This brings up the Revision History panel, as shown in Figure 2-13.

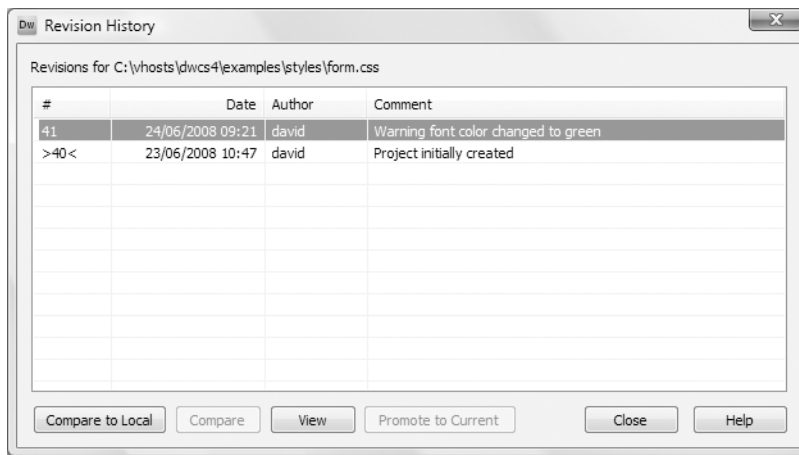


Figure 2-13. The Revision History panel lets you inspect different versions of the file in the repository.

The Revision History panel lists all versions of the selected file in reverse order, with the most recent at the top. When you select a file in the list, the buttons at the bottom of the panel perform the following actions:

- **Compare to Local:** This launches a third-party file-comparison utility to compare the selected version in the repository with your local working copy. See the “Using a file-comparison utility” section later in this chapter.
- **Compare:** This uses a third-party comparison utility to compare different versions in the repository. It is grayed out when only one version is selected in the panel.
- **View:** This opens the selected revision in the Document window. Dreamweaver gives the file a name based on its original file name and revision number. For example, revision 41 of `form.css` is opened as `form_rev41.css`.
- **Promote to Current:** This is how you roll back to an older version: by promoting the selected version to the most recent.

If Dreamweaver fails to commit your changes to the repository because of a conflict, you need to use this panel to get a copy of the latest version and resolve the conflict manually before committing the file back to the repository.

Keeping your working copies up-to-date



When using more than one computer or working in a team environment, you must always check out the latest version from the repository before making any changes to a file. You can do this by selecting the file in the Files panel and clicking the Check Out File(s) icon (see alongside) or by right-clicking and selecting Version Control ► Get Latest Versions from the context menu.

If a new file has been created by another member of the team, you can retrieve it from the repository by selecting Repository view from the drop-down menu at the top right of the Files panel. Highlight the file(s) that you want and click the Check Out File(s) icon or use the context menu. Switch the Files panel drop-down menu back to Local view when you have finished.

Color provides a useful visual clue as to what is displayed in the Files panel. When in Local view, folder icons are displayed in green. They're displayed in yellow when in Remote view or Repository view, and in red when you select Testing server from the drop-down menu.

Locking files

You can lock files in the repository to prevent others from editing them. Select the file(s) in the Files panel, right-click, and select Version Control ► Lock from the context menu. To unlock a file, select Unlock from the context menu.

Subversion also uses working copy locks (see “The three meanings of ‘lock’” in *Version Control with Subversion* at <http://svnbook.red-bean.com/> for an explanation). If you get error messages about working copy locks, select the file in the Files panel, right-click, and select Version Control ► Clean Up from the context menu.

Managing Dreamweaver sites

To change any settings in your site definition, select Manage Sites from the Site menu to open the Manage Sites dialog box (see Figure 2-14). Select the name of the site that you want to change and click Edit. This reopens the Site Definition dialog box, ready for you to update the settings. If you're feeling really impatient, though, the quickest way of opening the Site Definition dialog box is to double-click the site's name in the drop-down menu at the top left of the Files panel.

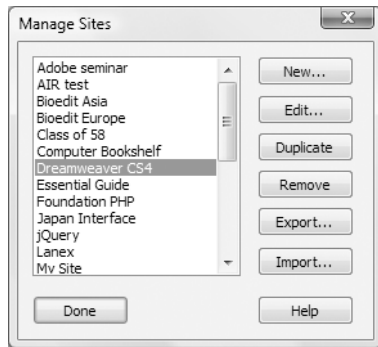


Figure 2-14.
The Manage Sites dialog box helps you organize the sites on your computer.

The other buttons on the right side of the Manage Sites dialog box are fairly self-explanatory. However, the following is a quick guide to each one:

- **New:** This offers two options: Site and FTP & RDS Server. The first opens the Site Definition dialog box. The second option is rarely used, but lets you create a direct FTP connection to a remote site (RDS is for ColdFusion only). You might want to use this to upload a single file without defining a local site in Dreamweaver.
- **Duplicate:** This creates an exact copy of the site definition for whichever site is highlighted in the left panel. You might find this useful if a new site shares common settings with an existing one. It's important to understand that creating a new site definition doesn't make a mirror version of the common files and folders. Editing or deleting a shared file in one site affects both sites, as there is only one set of files.
- **Remove:** This removes only the site definition from Dreamweaver. The actual files and folders remain untouched.
- **Export:** This exports your site definition as an XML file (Dreamweaver gives it an `.ste` file name extension). You can export multiple site definitions by using Shift-click or Ctrl/Cmd-click to select several sites in the left panel. If any of the site definitions contain login details for a remote server, Dreamweaver asks if you want to back up your settings with details of your username and password. If you select Share settings with other users, the login details are omitted. The export option you choose applies to all sites being exported at the same time. Dreamweaver then asks where to save the file. Just browse to the folder where you want to store the `.ste` files and accept the default value for File name. Definitions for all selected sites are exported in a single operation.
- **Import:** This imports site definitions from `.ste` files. If the `.ste` files are in the same folder, you can import multiple sites simultaneously. If a site of the same name already exists, Dreamweaver creates a duplicate site definition with a number after the name, rather than overwriting the existing definition.

Dreamweaver occasionally loses all your site definitions. It doesn't happen very often, but once is enough. Use the Export button to create a backup each time you add a new site or amend a site's definition. It could save you a lot of agony.

Setting options that apply to all sites

The options set in the Site Definition dialog box are specific to each individual site. Any edits that you make to the settings apply immediately to that site, but not to any other. Before concluding this chapter, let's take a look at Dreamweaver preferences that affect all sites. All are accessed through the Preferences panel, which is on the Edit menu in Windows and the Dreamweaver menu on a Mac. Alternatively, use the keyboard shortcut, Ctrl+U/Cmd+U.

Setting new document preferences

The New Document category of the Preferences panel, shown in Figure 2-15, sets global preferences for all web pages created in Dreamweaver.

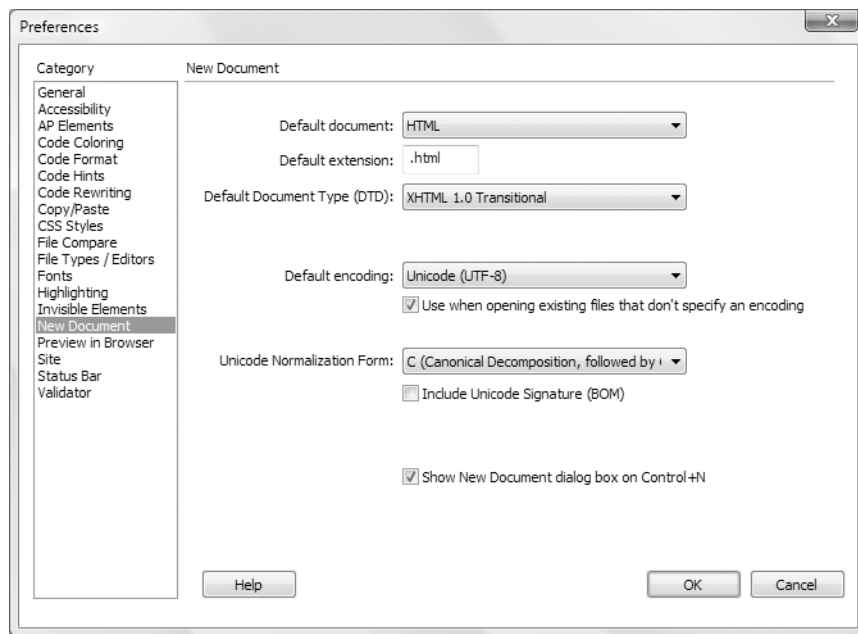


Figure 2-15. The New Document category of the Preferences panel

Let's take a look at what each of the options means:

- **Default document:** This lets you to choose the type of document that will be created when you use the keyboard shortcut for a new document (Ctrl+N/Cmd+N). For this to work, you must deselect the option at the bottom labeled Show New Document dialog box on Control+N/Cmd+N. Otherwise, the dialog box shown in Figure 3-2 will appear.
- **Default extension:** This affects only HTML files. Change the value only if you want to use .htm to maintain unity with the naming convention of older sites.

- **Default Document Type (DTD):** This sets the default DOCTYPE declaration for all new web pages. You cannot set one value for, say, .html and another for .php pages. See the next section for more information about choosing a default document type.
- **Default encoding:** This lets you choose the character set to be used in all web pages. The Dreamweaver default is Unicode (UTF-8). In the Mac version, this is listed as Unicode 5.0 UTF-8. The checkbox below this option tells Dreamweaver to use the same character set to display existing pages that don't specify a particular encoding. It doesn't insert any extra coding in such pages.
- **Unicode Normalization Form:** This is required only when using UTF-8 for encoding. It should normally be set to C (Canonical Decomposition, followed by Canonical Composition), and the Include Unicode Signature (BOM) checkbox should be deselected. If you use any other encoding, set Unicode Normalization Form to None.

Choosing the default document type

Many people misunderstand the purpose of the DTD (the DOCTYPE declaration before the opening `<html>` tag). It simply tells the browser how you have coded your page and is intended to speed up the correct rendering of your design. It's not a badge of honor or magic spell that somehow renders your web pages future-proof, although omitting the DOCTYPE altogether or using the wrong one switches most browsers into quirks mode, which could result in your page not looking the way you expect. The default setting is XHTML 1.0 Transitional, and this is the appropriate choice for most people when creating a new web page *as long as you understand the stricter rules imposed by Extensible Hypertext Markup Language (XHTML)*.

Visit <http://www.w3.org/TR/xhtml1/#diffs> to learn about the differences between HTML and XHTML.

The full range of options is as follows:

- **None:** Don't use this. All pages should have a DOCTYPE declaration.
- **HTML 4.01 Transitional:** Choose this if you don't want to use XHTML.
- **HTML 4.01 Strict:** This excludes deprecated elements (those destined for eventual elimination). Use this only if you have a good knowledge of HTML and have made a conscious decision not to use XHTML.
- **XHTML 1.0 Transitional:** This offers the same flexibility as HTML 4.01 Transitional by permitting the use of deprecated elements but applies the stricter rules of XML.
- **XHTML 1.0 Strict:** This excludes all deprecated elements. Use this only if you are competent with XHTML.
- **XHTML 1.1:** This DTD should not be used on pages delivered using the text/html MIME type, the current standard for web servers.
- **XHTML Mobile 1.0:** This is a subset of XHTML Basic for mobile devices. You can find the full specification at <http://www.openmobilealliance.org/tech/affiliates/wap/wap-277-xhtmlmp-20011029-a.pdf>.

If you choose an HTML document type, Dreamweaver automatically creates code according to the HTML specification. Similarly, if you choose XHTML, your code automatically follows the stricter rules, using lowercase for tag names and event handlers and inserting a closing slash in empty tags such as ``. You need to be careful when copying and pasting code from other sources. If you're not sure about the quality of the code, run Commands ► Clean Up XHTML, which should correct most, if not all, problems.

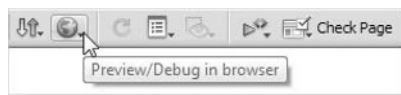
If you select a Strict DTD, it's important to realize that Dreamweaver does *not* prevent you from using deprecated elements or attributes. Dreamweaver expects you to understand the difference yourself.

Setting options for Preview in Browser

Live view, which was described in the previous chapter, is very useful for getting a quick view of what your page is likely to look like in a standards-compliant browser, but it's no substitute for viewing it in your main target browsers. Pressing F12/Opt+F12 or using the menu option File ► Preview in Browser automatically launches your default browser and displays the page currently open in the Document window.

Dreamweaver normally detects your default browser the first time that you use this option, but you can also designate other browsers in the Preview in Browser category of the Preferences panel. If Dreamweaver has detected other browsers on your system, they are listed in the Browsers field. You can designate one of them as your secondary browser, which can be launched using Ctrl+F12/Cmd+F12 as a shortcut.

Add other browsers by clicking the plus (+) button. Type the browser's name in the Name field, click the Browse button to locate its executable file, and then click OK to register it. The Edit button lets you change the details of the selected browser. Click the minus (-) button to remove the selected browser from the list. Although default keyboard shortcuts exist for only two browsers, you can launch the current page in one of the other browsers by using File ► Preview in Browser or clicking the Preview/Debug in browser icon on the Document toolbar, as shown in the following screenshot.



The Preferences panel offers the option of previewing pages using a temporary file. By default, this option is disabled. Opinions differ on whether to enable it. The advantage of using a temporary file for previewing is that you don't need to save the file beforehand, so you can test various things without needing to overwrite the existing file. On the other hand, you cannot test server behaviors that insert, update, or delete database records with a temporary file. So, in a local testing environment, my preference is to leave the option disabled. However, if you have defined your remote server as the testing server,

the opposite advice is true. You should enable preview with a temporary file for the following reasons:

- Even if you haven't entered any details in the Remote Info category of the Site Definition dialog box, Dreamweaver uploads the file to the remote server and permanently overwrites the existing file on the remote server.
- Dependent files, such as images, style sheets, and external JavaScript files, must also be uploaded to the remote server unless you preview using a temporary file.

Using a file-comparison utility

Dreamweaver lets you specify a third-party application to compare files. It can be used with files in your local site, on the remote server, or in a Subversion repository.

Setting up the File Compare feature

If you already have a file-comparison utility installed on your computer, all that's necessary is to register the program inside the Dreamweaver Preferences panel. If you don't yet have one, here are some suggestions:

- For Windows, consider WinMerge (<http://winmerge.sourceforge.net/>) or Beyond Compare (<http://www.scootersoftware.com>). WinMerge is free. Beyond Compare is moderately priced, but you can try it free for 30 days.
- For Mac OS X, you might use TextWrangler or BBEdit (both from <http://www.barebones.com>). TextWrangler is not just a file-comparison utility; it's an excellent script editor, and it's free. BBEdit is expensive if you only need it for file comparison, but it is widely recognized as the Rolls Royce of Mac script editors.

Once you have installed a file-comparison utility, select File Compare in the Preferences panel, click the Browse button, and navigate to the executable file for the program. Windows users should have little difficulty recognizing the correct file to select; it will normally be in a subfolder of Program Files.

On a Mac, the location is somewhere you may never even have known existed:

- **TextWrangler:** Macintosh HD:usr:bin:twdiff
- **BBEdit:** Macintosh HD:usr:bin:bbdiff (this is the BBEdit file-comparison utility—make sure you choose bbdiff and not bbedit, which is listed just below it)

Even though the `usr:bin` directory is normally hidden on a Mac, the Dreamweaver Select External Editor dialog box will display it by default. All you need to do is select the correct file name and click Open. If you can't find `twdiff` or `bbdiff`, open Preferences from the TextWrangler or BBEdit menu, select Tools, and click the Install Command Line Tools button.

Comparing files

To compare two files in the same site, highlight both in the Files panel, right-click, and select Compare Local Files.

If you select just one file in the Files panel and right-click, the context menu will display either Compare with Remote or Compare with Local, depending on the location of the selected file. For this type of comparison, Dreamweaver will select only a file of the same name and in the same location on the other computer. So you can compare the local or remote equivalent of `myfile.php` in `myfolder` in the same Dreamweaver site but not `myotherfile.php` or the same file in a different folder or different site.

Beyond Compare produces a false negative when comparing the remote and local versions of a file. This is easily remedied by opening the main Beyond Compare window and selecting Tools ► Options ► Startup. Set Show dialog with quick comparison results to Rules-based quick compare.

You cannot use the merge or copy feature of your file-comparison program to make changes to a remote file, because Dreamweaver works with a temporary copy of the remote file rather than the original. Local files can be changed, because you always work with the original.

On Windows, you can merge local and remote versions of a file by launching Beyond Compare outside Dreamweaver. Select New from the Beyond Compare Session menu, and choose Synchronize with FTP site. Alternatively, make a copy of the local file in the Files panel and give it a different name. You can then download the remote file and compare both of them locally.

Chapter review

This chapter has covered a lot of ground. If you're an experienced developer, you'll probably need to implement most of the features. If you're just setting out and learning how to use Dreamweaver, the most important sections were "Deciding where to test your pages" and "Creating the site definition."

As long as you have defined your site correctly in Dreamweaver, you should be on solid ground. But if you run into problems later with internal links not pointing to the right location or updating correctly, the answer almost certainly lies in this chapter. You can change the settings for a site any time by opening the Site Definition dialog box as described in the "Managing Dreamweaver sites" section. Site definition is the key to success in Dreamweaver. Leave the more advanced options, such as Subversion integration, until after you have gained more experience.

In the next chapter, you'll get down to creating web pages with the main focus on creating modern, standards-compliant layouts using CSS.