

Chapter 1

Becoming Familiar with SharePoint Designer

The Internet has long been one of the favorite mechanisms of expression because of the wide reach, connection, and exposure it offers. It's one of the basic means of communication in the 21st century and has drawn people closer in unique ways. Having a presence on the Internet is a pivotal requirement for any organization, irrespective of its size, nature, or range of operations. Web sites on the Internet provide the canvas that organizations can use to explain their missions and goals, advertise their products, perform business transactions, and communicate with their customers and partners.

It's apparent that the Internet as a medium offers tremendous prospects and opportunities. To exploit this medium, Web site designers have a range of Internet technologies to choose from. From simple markup languages (such as HTML) to complex application development technologies (such as ASP.NET), there are a variety of platforms on which you can base your Web site application. To achieve the most from these technologies without having to re-create a lot of work, many Web site development tools and products are available to you. Microsoft's key offerings for these tools and products have been FrontPage and Visual Studio. While Visual Studio is targeted to Web developers and complex Web application development, FrontPage is designed to provide a simpler no-code-based software development tool that helps Web site designers focus on designing Web sites rather than have to deal with the complex code that goes behind Web site development and still be able to create complex Web sites.

FrontPage offers Web site designers a number of components and What You See Is What You Get (WYSIWYG) features that help in building important sections of Web sites. It provides wizard-based interfaces that aid in developing Web pages without having to write a lot of code behind the Web pages and creating key Web site features, such as site navigation, data collection

IN THIS CHAPTER

Introducing SharePoint Designer

Understanding SharePoint Designer basics

Exploring hidden metadata and raw Webs

Maintaining Web sites

forms, dynamic effects, and graphics and animation. Along with that, it offers a number of Web site management features that provide for ease in managing Web sites as well as importing, exporting, and publishing content to Web servers.

With the advent of SharePoint technologies, FrontPage has been replaced primarily because of the need to address the requirements of Web site designers in the SharePoint and non-SharePoint environments. FrontPage 2003 is the last version of FrontPage, and FrontPage is now available as two separate products: Expression Web (for non-SharePoint development) and SharePoint Designer (for SharePoint development). While SharePoint Designer falls into the Microsoft Office suite (although it's not part of any Office 2007 suite and is sold separately), Expression Web is part of the new Expression Studio suite of products, which includes Expression Blend, a program once called Interactive Designer.

In this chapter, I introduce you to SharePoint Designer and help familiarize you with some basics for developing Web pages and Web sites using SharePoint Designer. Although I try to keep this chapter simple, if you run across a term, feature, or component that sounds confusing, don't be concerned because I cover these elements in more depth throughout this book.

Introducing SharePoint Designer

SharePoint is one of the fastest-growing Microsoft products and offers a platform for developing enterprise-level Web applications that focus on the sharing and collaboration of content and documents. In its most basic of definitions, SharePoint provides the infrastructure for creating and maintaining Web sites that can be used for a variety of Internet-based operations besides sharing Office or non-Office content and collaboration. You can later customize these Web sites that SharePoint creates to suit your business needs, format, or branding.

SharePoint Designer is Microsoft's premier product for Web site designers who are tasked with customizing and branding SharePoint Web sites. Although SharePoint Designer in its current release can be used for non-SharePoint Web site development, its full capability is exposed when you work with SharePoint Web sites. However, before I discuss SharePoint Designer and its features, I want to briefly mention its predecessor, FrontPage, and the origins of SharePoint Designer.

Exploring the legacy of FrontPage technologies

Microsoft acquired FrontPage quite a few years ago from a company called Vermeer Technologies Incorporated (VTI, an acronym you encounter a lot when exploring the internals of SharePoint Designer later in this chapter). FrontPage's Web site authoring and management techniques were really ahead of their time and allowed Web site developers to easily create Web sites on local computers and then push them to Web servers or even work directly on the Web site residing on the Web server. The underlying authoring techniques of FrontPage were exploited by a lot of other Microsoft products, such as Visual InterDev, Office, and Visual Studio.

The concept was simple: The FrontPage program installed on the client computer provided the *client-side component* that could be used for creating Web pages and designing Web sites. The Web server was set up with a *server-side component* — namely, FrontPage Server Extensions (FPSE) — that would communicate with the client-side component for providing the basic infrastructure for three operations:

- Moving content from the client machine to the Web server
- Authoring or modifying content directly on the Web server
- Enabling certain Web components that provide functionality for Web sites

These authoring technologies that FrontPage introduced are still very much alive and used by SharePoint Designer, Expression Web, and Office 2007 products when working with SharePoint and non-SharePoint sites.

CROSS-REF For more on these authoring mechanisms and protocols, see Chapter 3.

FrontPage could be used to create Web pages and design Web sites by using a set of Web components and features offered with the product. These features included components such as hit counters, shared borders, link bars based on the navigation structure, and many more. The list of these components grew with each new version of FrontPage.

On the server side, FPSE provided the basis for the first SharePoint implementation offered with Office XP/2002. This release of SharePoint was called SharePoint Team Services (STS) and stemmed from a less-used ASP-based Office 2000 document management feature called Office Server Extensions (OSE). STS allowed developers to create Web sites by using Web site templates that were provided with the product. These Web sites provided basic collaboration and document management features, including document libraries and lists.

FrontPage XP/2002 became the client-side tool for customizing and modifying Web pages inside STS sites. FrontPage 2002 also provided for the interface that could be used to create new document libraries and lists (SharePoint content) as well as new SharePoint Web pages and Web sites, publishing STS sites from one Web server to another. All these SharePoint features were enabled in FrontPage 2002 only when an STS site was open in it and were not available for non-SharePoint sites.

This client-server relationship was thus established as the basis for the next versions of FrontPage and SharePoint. FrontPage 2003 was offered as the client-side designing tool for working with Windows SharePoint Services 2.0 (WSS v2). It could be used to develop SharePoint content and exploit the features offered by WSS v2 sites.

Following this relationship, SharePoint Designer was introduced as the designing tool for the current version of SharePoint, WSS v3, and Microsoft Office SharePoint Server (MOSS). Although SharePoint Designer is backward-compatible to a certain extent with the previous versions of SharePoint, its complete set of features are exposed only when using it with WSS v3 and MOSS sites.

Comparing SharePoint Designer with FrontPage 2003

As discussed earlier, SharePoint Designer is the designing tool for the current release of SharePoint. SharePoint Designer inherits many of its features from FrontPage 2003. Besides these features and enhancements, SharePoint Designer makes available a set of new technologies and features, such as workflows, page layouts, etc., that SharePoint v3 exposes.

While I highlight key improvements in SharePoint Designer as compared to FrontPage 2003 throughout this book, Table 1.1 compares some of the major features and components available in FrontPage 2003 and SharePoint Designer.

TABLE 1.1

Comparing SharePoint Designer with FrontPage 2003

Features	FrontPage 2003	SharePoint Designer
Task panes	Introduced in FrontPage 2003 for working with various features, such as layers, behaviors, layouts, etc.	Extensively uses dockable task panes to manage features, such as CSS, Tag Properties, Data Sources, Reports, etc. Property grids provide a Visual Studio–like interface for working with tag and control properties.
Cascading Style Sheets (CSS)	Interface available to create new styles and implement style sheets on Web pages	Brand-new intuitive CSS application and management interface using toolbars and task panes that can be used to create and manage styles. CSS layout templates also available.
Master pages	Not available	Ability to create, attach, and detach master pages and content regions. Reporting for master pages is also available
ASP.NET 2.0 controls	Not available	Available with .NET Framework 2.0 installed on the machine
Database wizards	Used to create Web pages for interfacing with databases	Deprecated and replaced by ASP.NET 2.0 data controls
SharePoint content	Can be used to create and manage content with WSS v2 sites. Limited backward-compatibility also available for STS sites	Can be used to create content for and manage WSS v2 and WSS v3 sites
Workflows	Not available in WSS v2	Workflow Designer available for creating and deploying declarative workflows for WSS v3 sites
Data views	Available for showing data from databases, SharePoint sources, XML files, etc.	Available for showing data from databases, SharePoint sources, XML files, etc. Extends the FrontPage 2003 capabilities by allowing for creation of forms that can be used to insert and update data besides showing it

Features	FrontPage 2003	SharePoint Designer
Web components	A number of FPSE- and WSS v2–based components available, such as hit counters, confirmation fields, etc.	Inherits these Web components from FrontPage 2003
Dynamic Web templates	Ability to create, attach, and detach dynamic Web templates and manage editable regions	Inherits this feature from FrontPage 2003
Data forms	Not available. List forms can be modified to a certain extent by using Collaborative Application Markup Language (CAML) features.	Can be used to create advanced data forms for SharePoint list and document libraries, databases, etc., using the new and advanced Data Form Web Part
Reporting for Web sites	Site and usage reporting available depending on the server setup	Advanced CSS error and usage reporting, accessibility, and compatibility reporting available using task panes
Code and design features	IntelliSense and auto code completion available for HTML and client-side scripts	Visual and formatting marks for HTML tags and ASP.NET 2.0 controls in the Design view. IntelliSense for HTML, JavaScript, and ASP.NET 2.0
Data sources	SharePoint data sources available through the Data Source Library task pane	SharePoint and non-SharePoint data sources available through the Data Source Library task pane. Aggregate data sources available for linking data sources together
Re-ghosting SharePoint pages	Not available	User interface available to selectively re-ghost SharePoint Web pages

Some of the differences shown throughout this book are apparent; SharePoint Designer is the newer version and exploits many features that the new SharePoint 3.0 itself has to offer. Workflows, for example, is a new technology that SharePoint 3.0 exposes to SharePoint Designer for creating advanced logic-based processes that could help in implementing common business actions and purposes. The ability to use ASP.NET 2.0 controls through a wizard-based approach is also new for SharePoint Designer, which now offers a simpler Visual Studio/Visual InterDev–like interface to work with ASP.NET 2.0 control properties. Some features, such as the Database Results Wizard and shared borders, are deprecated and have been replaced in favor of newer and better technologies. Throughout this book, I talk about these differences while also discussing the new features.

Although SharePoint Designer offers its full capabilities for SharePoint sites, you could be a Web site designer concentrating totally on non-SharePoint Web sites. In that case, the SharePoint features of SharePoint Designer might just be useless to you until you start developing for SharePoint, and you might want to invest in a trimmed-down version of SharePoint Designer, such as the current release of Expression Web, which is a subset of SharePoint Designer.

Choosing between Expression Web and SharePoint Designer

This choice is quite simple: If you as a Web site designer need to concentrate primarily on SharePoint sites, the obvious choice is SharePoint Designer. However, if you don't develop content for SharePoint sites, you can invest in Expression Web. Expression Web offers almost all the non-SharePoint features of SharePoint Designer, such as ASP.NET 2.0 controls, master pages, and CSS, but has the limitation that it can't open SharePoint sites. It lacks all the SharePoint capabilities that SharePoint Designer offers. So, if a non-SharePoint Web server hosts your Web site and you have no plans to develop SharePoint content, Expression Web can provide you with all the tools you need for your Web site development. The non-SharePoint interface for the current release of Expression Web and SharePoint Designer is essentially similar and thus you can also use this book as a reference for Expression Web.

Understanding SharePoint Designer Basics

In this section, I discuss some basic concepts and terminology about Web site designing from the SharePoint Designer perspective, helping you feel more comfortable as you read the rest of this book. Most of the concepts surrounding SharePoint Designer are based on FrontPage technology, so if you're already familiar with FrontPage 2003, this might just emphasize your knowledge about SharePoint Designer.

Meeting the requirements

Although the basic SharePoint Designer setup is fairly simple and doesn't require much user interaction, you might want to have these prerequisites set up on your computer before you install SharePoint Designer so that all its features are readily available to you:

- .NET Framework 2.0
- .NET Framework 3.0

You can download these from the Microsoft Web site at www.microsoft.com/downloads.

Having these tools downloaded to your computer prior to installing SharePoint Designer ensures that you can properly use the elaborate ASP.NET 2.0 features of SharePoint Designer. But you can also download them after installing SharePoint Designer.

Prior to installing SharePoint Designer, you also need to ensure that your machine meets these basic system requirements:

- **Processor:** 700 megahertz (MHz) processor or higher
- **Memory:** 512 megabyte (MB) RAM or higher

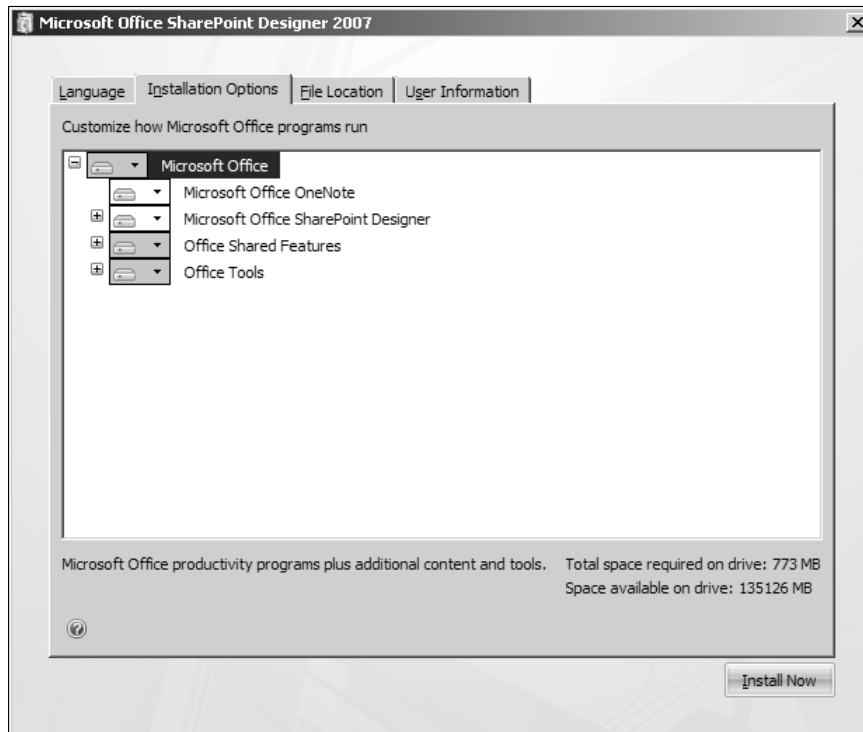
- **Hard drive:** 1.5 gigabyte (GB); a portion of this disk space is freed after installation if you remove the original download package from the hard drive.
- **Drive:** CD-ROM or DVD drive
- **Display:** 1024 × 768 or higher resolution monitor
- **Operating system:** Microsoft Windows XP with Service Pack (SP) 2, Windows Server 2003 with SP1, or later operating system

If you work with SharePoint Designer on SharePoint sites, then you need to have a Web server set up for WSS or MOSS.

The setup process for SharePoint Designer is fairly simple and is similar to the Office 2007 setup process. The Setup Wizard allows you to choose to either set up the basic installation by using the Install Now button or customize the setup using the Customize button. The Customize button, as shown in Figure 1.1, allows you to change the installation directory for SharePoint Designer and pick features that you want to install during setup.

FIGURE 1.1

Customization options during setup for SharePoint Designer



After you install SharePoint Designer on your computer, choose Start ⇨ Control Panel ⇨ Add or Remove Programs to change (add or remove features), repair, or remove SharePoint Designer. On Windows Vista machines, this is accessible by choosing Start ⇨ Control Panel ⇨ Programs and Features. The first time you open SharePoint Designer, you're asked whether you want to make SharePoint Designer your default program for Web site development. If you choose to use SharePoint Designer as the default program, it's set up so that Web content (HTML or ASP.NET pages, style sheets, etc.) on your computer opens with SharePoint Designer.

TIP

Setup administrators might want to skip the introductory messages that are shown the first time SharePoint Designer opens and allow their users to directly start using SharePoint Designer rather than having to make these choices. You can learn more about turning off these messages at <http://support.microsoft.com/kb/929767>.

The SharePoint Designer setup can also be configured using Office policy templates, which allow administrators to control the features available within SharePoint Designer to their users.

NOTE

The Microsoft support article at the following Web site provides information about how administrators can use Microsoft Office templates with Microsoft Office 2007 programs (including SharePoint Designer): <http://support.microsoft.com/kb/924617>.

After your SharePoint Designer setup is complete, you might also want to install the SharePoint Designer Service Pack (SP) 1 from the Microsoft Web site at www.microsoft.com/downloads.

Understanding the structure of Web pages and Web sites

The structure of Web pages in general is pretty flexible and spans to the imagination of a Web site designer. As a designer, you have the power to decide how you want to lay out your Web pages. SharePoint Designer provides you with templates that you can use for starting up.

CROSS-REF

For more on working with basic Web page components, see Chapter 5.

From the SharePoint Designer perspective, it's important to understand that a Web page essentially has a Design view and a Code view. The Code view shows the code (HTML, JavaScript, ASP.NET code, etc.) of the Web page and allows direct modification by using features such as IntelliSense and auto code completion. When you click the Design view tab of a Web page, SharePoint Designer renders the code associated with the Web page into a WYSIWYG format that assists in designing and previewing the page layout. Some files, such as style sheets, don't need a Design view, so only a Code view is displayed for them.

SharePoint Designer's complex rendering logic also renders ASP.NET 2.0 controls, the master page application, and the style sheet application on Web pages. While rendering Web pages for the Design view, SharePoint Designer also stores the properties and attributes of the tags and controls present in the code of the Web page. These properties are then presented in the various task panes for modification.

NOTE

All task panes are accessible under the Task Panes menu.

CROSS-REF

For more on property grids and task panes, see Chapters 4 and 6.

I now take you through the steps for creating a simple Web site and then adding some content to that Web site.

One important thing to understand while using SharePoint Designer to create Web pages is that Web pages by nature reside inside Web sites. Although you can save your Web pages wherever you want on your computer, it's always best that you create a folder for your Web site and store all Web pages and other content inside that folder. Such a folder is called the root folder of your Web site. Creating a root folder assists you in better management of Web pages and other content that you create during the course of building a Web site:

1. **Create a root folder for your Web site at the location of your choice on your computer.** This ensures that you remember the location where your Web pages and content are stored. Ensure that your root folder is empty.
2. **Open SharePoint Designer and then choose File↔New.** The New dialog box opens. This dialog box is new to SharePoint Designer and provides a consolidated interface to create new Web content (pages, JavaScript file, style sheets, etc.).

CROSS-REF

For more on the New dialog box, see Chapter 4.

3. **Click the Web Site tab.** This tab allows you to choose from a list of Web site templates available in SharePoint Designer. For this exercise, use the One Page Web Site template.
4. **Click Browse, which is next to the Specify the location of the Web site dropdown menu.** Using the New Web Site Location dialog box, specify the location of the root folder that you created in step 1.
5. **Click OK.** SharePoint Designer opens your Web site at the root folder and automatically places a `default.htm` file (because you selected the One Page Web Site template) inside the root folder. The folder is now listed in the Folder List task pane (which can be toggled on or off by pressing `Alt+F1`), which is displayed in the left corner of the SharePoint Designer window by default.

After following these steps, the icon for your root folder inside Windows Explorer changes to a folder icon with a globe on it, indicating that it's a Web site (a disk-based Web site). SharePoint Designer Web sites are basically one of two types:

- **Disk-based:** The content of this Web site resides inside a folder on your local computer. You work on the disk-based Web site locally and then publish the content you create locally to a Web server-based location that might reside either on your local computer (if your computer is a Web server itself) or a remote Web server computer (a site-hosting service provider, for example). The folder icon with a globe on it (an `epdbw.ico` file stored in the `_vti_pvt` folder) indicates this type of Web site.

- **Server-based:** The content of this Web site is located on a remote Web server and is accessible to the SharePoint Designer client for modification by using a remote-authoring mechanism, such as FPSE, SharePoint, FTP, or WebDAV. SharePoint Designer now allows for remote authoring (also called live editing) of server-based FTP and WebDAV sites. SharePoint sites also fall into this category of Web sites.

When you open the root folder of your Web site, SharePoint Designer has created a number of other files and folders inside the root folder.

NOTE

Many of these files and folders are hidden by default, so you have to enable the viewing of hidden folders in Windows Explorer by choosing **Tools** ⇨ **Folder Options** ⇨ **View**. Click the **Show hidden files and folders** radio button in the **Advanced Settings** list.

The two folders that result from the previous steps are the `_vti_cnf` and the `_vti_pvt` folders. The reason for having these folders is that they provide the underlying infrastructure files for many features that SharePoint Designer exposes. For example, in the `_vti_pvt` folder is a file called `structure.cnf`. This file stores the navigation structure of the Web site, which can be created by using the Navigation pane in SharePoint Designer. So, essentially, if someone deletes this file or folder using Windows Explorer, the navigation structure of your Web site is lost, and any Web components based on the navigation structure are useless.

CROSS-REF

For more on the Navigation pane, see Chapter 14.

NOTE

The folders that have the prefix `_vti_` aren't displayed in SharePoint Designer. So, if you want to hide the contents of a folder from being edited using the SharePoint Designer interface, you can do so by adding the `_vti_` prefix to it.

The other important thing to understand about these folders is that they're used for the FPSE-based features of SharePoint Designer. Also, a number of other folders — such as `_derived`, `_overlay`, and `_themes` — are created in the root folder when you use FPSE features. If you delete these folders, any FPSE-based components that depend on these folders and the files inside them stop working.

Exploring Hidden Metadata and Raw Webs

Files and folders inside the root folder of a SharePoint Designer Web site are collectively called the hidden metadata of the Web site. The metadata inside the root folder keeps track of whether the Web site is a disk-based Web site or a server-based Web site. It also stores information about the list of subsites that the root site might contain. The metadata is a vital resource for performing a number of site management operations, such as hyperlink recalculation and creating the navigation structure.

If you decide to move away from using the FPSE features of SharePoint Designer, you might not need this metadata and can configure SharePoint Designer to not create the hidden metadata. A SharePoint Designer Web site that doesn't have metadata in it is called a raw Web. To convert your Web site to a raw Web, follow these steps:

1. In SharePoint Designer, open the Web site you created earlier in this chapter by choosing **File** ⇨ **Open Site**. The Open Site dialog box opens, with the Web Sites pane selected in the left Navigation pane. The Web Sites pane shows the list of Web sites that you've created.
2. Choose your Web site and then click **Open**.
3. Choose **Site** ⇨ **Site Settings**. The Site Setting dialog box opens.
4. Deselect the **Manage the Web site using hidden metadata files check box**. As indicated in the explanation text next to this check box, SharePoint Designer adds hidden metadata files to your root folder to enable and maintain certain features.
5. Click **OK**. A dialog box asks you for confirmation. The warning message indicates that if you elect to not use hidden metadata, you lose the navigation of your Web site and many FrontPage Web components stop being updated.
6. Click **Yes if you want to continue**.

After you deselect this option, the root folder of your Web site no longer has any `_vti` folders in it. This means that SharePoint Designer now has no ability to track FrontPage Web components and features. These features are thus disabled from the SharePoint Designer user interface when this option is deselected. For example, if you choose **Insert** ⇨ **Web Component** and then open the **Insert Web Component** dialog box, most of the features are disabled when you deselect the hidden metadata usage check box. Also, the **Recalculate Hyperlinks** menu option is no longer available in the **Site** menu.

CROSS-REF For more on **Recalculate Hyperlinks**, see **Chapter 23**.

If you later need to use the SharePoint Designer features that rely on the hidden metadata files, you can click the **Manage the Web site using hidden metadata files check box** again. SharePoint Designer then adds new hidden metadata to your Web site and starts using it for maintaining cross-page dependencies and other FrontPage Web components.

CAUTION Any metadata that's removed by deselecting the hidden metadata check box is completely lost and isn't recoverable. When you click the check box again, new metadata is created, and you have to rebuild the components.

The ability to remove hidden metadata has long been asked for by Web site developers who don't use FrontPage-based components and want to see cleaner Web site content directories. Because SharePoint Designer now offers the capability to use ASP.NET 2.0 controls and features extensively, most of the FrontPage legacy functionality can be obtained by newer, advanced ASP.NET 2.0 controls. Raw Webs provide these advantages for Web site developers:

- Developers now have a mechanism to ensure that they don't inadvertently use any FrontPage legacy features.
- Because SharePoint Designer now offers the capability of live editing using FTP and WebDAV, developers can choose not to use FPSE as a remote-authoring mechanism.
- They don't have to manage the hidden metadata when moving Web content from one location to another.

Creating basic Web pages

Continuing with efforts to understand the basics involved in Web content development, I take you through some steps that you generally follow in SharePoint Designer when creating Web pages. I want you to become familiar with the SharePoint Designer terminology in this chapter so that you don't feel uncomfortable when you delve into more complicated Web site designing concepts.

Follow these steps to create simple Web pages using various templates that SharePoint Designer offers:

1. **In SharePoint Designer, open the disk-based Web site you created earlier in this chapter.** Ensure that the Manage the Web site using hidden metadata files check box is selected.
2. **Choose File ⇨ New to open the New dialog box.**
3. **Click HTML under the Page tab to create a blank HTML Web page and then click OK.** The Design view of the newly created HTML page is shown.
4. **Choose File ⇨ Save or press Ctrl+S to save the page.** The Save As dialog box opens, with your currently opened Web site listed in the Save dropdown menu.
5. **Using the File name text field, type a name for your Web page and then click Save.** HTML pages have the extension `.htm` or `.html`. You can save the Web page by using either extension.
6. **Choose File ⇨ New to open the New dialog box.** You can now create a new CSS from the existing templates in SharePoint Designer.

CROSS-REF

For more on Cascading Style Sheets (CSS), see Chapter 12.

7. **Click Style Sheets and then click the Downtown style sheet template.** A new CSS based on the Downtown template opens. This template appears in the Code view.
8. **Press Ctrl+S to open the Save As dialog box and then save the style sheet with the name of your choice.** Style sheets have the extension `.css`, so make sure that you keep this extension when saving a style sheet. Because you're saving these files into the Web site's root folder, the newly created Web page and the style sheet are listed in the Folder List task pane.
9. **Using the Folder List task pane, double-click the Web page you created in the previous steps to open it in SharePoint Designer.**
10. **Choose Format ⇨ CSS Styles ⇨ Attach Style Sheet to open the Attach Style Sheet dialog box.** You use this dialog box to attach to the Web page the style sheet you previously created.
11. **Click Browse next to the URL text field to open the Select Style Sheet dialog box, click the style sheet file you created in the previous steps, and then click Open.** Ensure that the Current Page radio button is selected so that the style sheet applies only to the current page.
12. **Click OK to apply the style sheet to the Web page.**

The look and feel of the Web page changes after the application of the style sheet. The Web page now has a new background, and if you type some text on the Web page, the color of the text is white. The SharePoint Designer Design view renders the Web page in the Design view by combining the formatting implemented in the style sheet attached to it.

The New dialog box that you worked with in the previous exercise is the interface that consolidates the creation of all Web content within SharePoint Designer. This is different from FrontPage 2003 and the previous versions where page and Web content creation were scattered around in different task panes. Having a single interface for creating all SharePoint and non-SharePoint content is advantageous because it eliminates the need to search around the user interface for developing content.

The New dialog box can be used to create HTML and ASPX Web pages, JavaScript files, CSS, master pages and dynamic Web templates (DWTs), XML, and TXT files. Also, it offers the Web Site tab, which can be used to create Web sites (SharePoint and non-SharePoint) based on templates.

NOTE

If you open a SharePoint site in SharePoint Designer, the New dialog box also shows a tab called SharePoint Content. This tab can be used to create SharePoint-based content, such as lists, document libraries, workflows, and page layouts. Also, for SharePoint sites, the Web Site tab lists a number of templates that you can choose from for your new SharePoint sites.

Apart from using the New dialog box, you can also use the Folder List task pane to create new Web pages and content. If you right-click in the empty area inside the Folder List task pane, a popup menu appears that has the New menu option that allows for the creation of Web pages and CSS. You can use this menu to create new folders and subsites.

Follow these steps to use the existing frame templates that SharePoint Designer provides to create a set of Web pages:

CROSS-REF

For more on frames, see Chapter 5.

- 1. In SharePoint Designer, open your Web site and then choose File ⇨ New to open the New dialog box.**
- 2. In the Page tab, click Frames Pages and then click the Header, Footer and Contents template.** As indicated in the description text, the template creates header and footer frames for navigation.
- 3. Click OK.** This creates a frameset page, as shown in Figure 1.2, that's displayed in the Design view in SharePoint Designer.
- 4. Save this page inside the root folder of your Web site.**
- 5. Click the New Page button on all the four frames inside the frameset page.** This creates new pages that fill up the frames present in the frameset page. Remember the order in which you clicked the New Page buttons. This is required to name the pages in the next step to keep track of which page is displayed in which frame.

FIGURE 1.2

A frameset page in the SharePoint Designer Design view



6. Press **Ctrl+S**, and you're asked to save all the HTML pages inside the frames one after the other in the order they were created. Because there are four frames, you have four new Web pages in the root after you save the frames pages. You can name the page for the left frame as `Left.htm`, the top frame as `Top.htm`, etc.

Your frameset page is now ready for the content that you want to put on the frames pages. For example, you could use the top frame page for a banner picture, and the left frame could be used to create hyperlinks that switch pages in the Contents frame of the frameset.

Right now, I want to divert your attention to the newly created pages from the previous exercise. The Folder List task pane lists these newly created Web pages in the root folder of your Web site. If you right-click on a Web page in the Folder List task pane, a popup menu opens with these options:

- **Open:** Opens the Web page in the SharePoint Designer Design view (or Code view)
- **Open With:** Allows you to choose the program that you want to open the Web page in
- **Open in New Window:** Opens the Web page in a new instance of SharePoint Designer
- **Set as Default Master Page:** If you've selected a master page, you can use this option to set the selected master page as the default master page for the Web site. If you don't have a master page selected, this option is grayed out.

CROSS-REF

For more on master pages, see Chapter 13.

- **New From Existing Page:** Creates a new page based on the selected page
- **Preview in Browser:** Opens the Web page in the default browser for previewing
- **Cut, Copy and Paste menu options:** Allows for moving Web pages from one folder location to another inside the Web site
- **Set as Home Page:** Sets the Web page as the default home page for the Web site
- **Rename:** Allows you to rename the Web page
- **Delete:** Deletes the Web page after requesting confirmation from the user
- **Publish Selected Files:** Opens the publishing page that can be used to publish the Web page to a new location. Web page publishing is discussed later in this chapter.
- **Don't Publish:** Marks the Web page so that it's not included when a publishing operation is performed on the Web site
- **Properties:** Opens the Properties dialog box, which can be used to view general properties of the Web page and choose categories (if hidden metadata is allowed) for the Web page

Options in this popup menu might change depending on the type of selection you make in the Folder List task pane and the type of Web site that you have open in SharePoint Designer. For example, if you have document checking enabled on a Web site (discussed later in this chapter), the Check Out menu option becomes available.

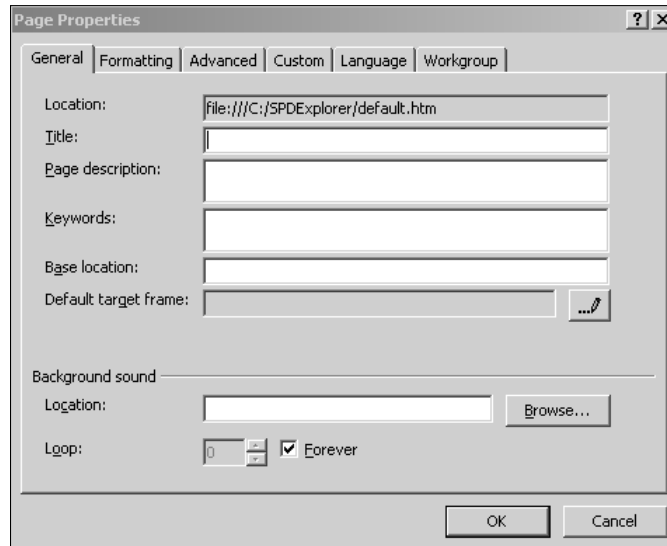
Right-click on the Web page after opening it in the Design view and then choose Page Properties from the popup menu. This opens the Page Properties dialog box, as shown in Figure 1.3, which can be used to set up a number of properties for the Web page. These properties include the title, background sound, background picture, etc.

The Page Properties dialog box has the following tabs that allow for setting a number of properties for a Web page. Most of these settings provide a user interface for applying HTML tags and attributes or style elements that define the look and feel of the Web page:

- **General:** Allows the user to set the <Title> tag, the <Base> tag, keywords, and description meta tags for the Web page. It also allows you to set the default target frame and background sound for a Web page.
- **Formatting:** Provides the interface to set the background picture or configure the background picture as a watermark on a Web page. This also allows for setting the background color, text color, and colors for hyperlinks.
- **Advanced:** Used to set margins for a Web page. This is done by applying the margin styles to the <Body> tag of the Web page.
- **Custom:** Allows for creating HTML meta tags, such as Refresh, Keywords, and Description, that define the behavior of the Web page or provide directions to the Web browser
- **Language:** Provides the interface to set the Content-Type and Content-Language meta tags for the Web page. These meta tags define the page's language and HTML encoding.
- **Workgroup:** Allows the Web site designer to create categories of Web pages and assign Web pages to those categories. These categories can then be used to create a categorized table of contents. This feature requires the hidden metadata to be enabled.

FIGURE 1.3

The Page Properties dialog box



Follow these steps to use the Page Properties dialog box to set some properties for the Web page that define the way the Web page looks and behaves:

1. Choose **File** ⇨ **New** to open the New dialog box and then create a new HTML Web page. The new page opens in the Design view.
2. Right-click on the Web page and then choose **Page Properties** from the popup menu. The Page Properties dialog box opens.
3. In the **General** tab, type the title for the Web page in the **Title** text field.
4. Click the **Formatting** tab, and in the **Colors** section, choose a background color for the Web page by using the **Background** dropdown menu.
5. Click the **Custom** tab. In the next steps, you use this tab to create a new system variable (an HTML meta tag) that refreshes the page every 5 seconds when viewed in a browser.
6. Click the **Add** button next to the **System variables** list.
7. In the **Name** text field, type **Refresh**, and in the **Value** text field, type **5** (for a refresh every 5 seconds).
8. Click **OK**. Your choices have been saved for this HTML page.
9. **Save the Web page**. You can now browse to the page by either right-clicking on it in the Folder List task pane and then choosing **Preview in Browser** from the popup menu or pressing **F12**.

The title you set for the Web page shows in the title bar of the Web browser. The background color of the Web page changes to the one you set. Also, the browser reloads the Web page every 5 seconds because of the `Refresh` meta tag that you created in the previous steps. The SharePoint Designer interface allows you to set the basic properties for a Web page without having much understanding of the code involved in the back end. Obviously, SharePoint Designer offers many other ways to manage these properties for Web pages, and I discuss those options throughout the rest of this book.

Working with simple Web sites

A Web site is basically a collection of resources that help realize the goals of an organization on the Internet. The resources for Web content are Web pages, pictures, supporting files (style sheets, JavaScript files, etc.) and programs (.exe, .dll, etc.) that contain logic and perform business operations. A Web site is accessible on the Internet through a Web server. The role of the Web server is to be a host for Web site content and process Web page requests made by Web browsers. A single Web server can host a large number of Web sites.

Now that you know how to create a simple Web site and add Web pages and style sheets to it, it's time to learn how to open and delete Web sites, create subsites, and establish general settings for Web sites inside SharePoint Designer.

Creating subsites within a Web site by using SharePoint Designer essentially involves the same steps you would follow to create a new Web site. That is, choosing **File** ⇨ **New** ⇨ **Web Site** allows you to pick the template you want to use for the subsite. However, just before you click **OK**, you need to click the **Add to current web site** check box. It's important to note that subsites inherit the server configuration of the parent Web site. This means that subsites inside SharePoint sites are always SharePoint subsites. Similarly, subsites inside FPSE parent sites are always FPSE subsites.

To open a Web site in SharePoint Designer, simply choose **File** ⇨ **Open Site**. The **Open Site** dialog box opens, with a number of panes that allow you to look for the Web site that you want to open. The **Web Sites** pane in the left **Navigation** pane shows the list of all Web sites that you've created or opened on the machine that has SharePoint Designer installed.

The **Current Site** pane, as the name suggests, is only available if you already have a site open in SharePoint Designer before you open the **Open Site** dialog box. It shows the list of folders and files for the Web site that's currently open and can be used to open subsites within the site. The **My SharePoint Sites** pane lists the SharePoint sites that you created or opened by using SharePoint Designer.

If you've previously opened the Web site in SharePoint Designer, try to see if you can find it in the recently opened list of Web sites by choosing **File** ⇨ **Recent Sites**.

NOTE

SharePoint Designer also maintains a list of recently opened sites and Web pages in the Windows registry at `C:\My Computer\HKEY_CURRENT_USER\Software\Microsoft\Office\12.0\SharePoint Designer`. SharePoint Designer uses this registry hive for saving user settings, including recently opened sites and Web pages, the default save location for Web sites, and the state of various SharePoint Designer user interface components.

If the Web site that you want to open doesn't already exist in the aforementioned lists, you can simply type the complete URL of the Web site in the Site Name text field. SharePoint Designer tries to establish a connection to the remote Web server and, if it finds the Web server hosting the Web site and can successfully authenticate to it, then opens the Web site for remote authoring.

CROSS-REF For more on remote authoring, see Chapter 3.

While choosing File ⇨ Open Site takes you to the Open Site dialog box, which allows you to open Web sites, choosing File ⇨ Open brings you to the Open File dialog box, which is used to open files and documents inside Web sites. If your page or document resides inside a Web site, SharePoint Designer opens the Web site itself. You can choose File ⇨ Close Site to close the Web site.

SharePoint Designer supports live editing and remote authoring for a number of Web server types. You can work directly on a Web site in SharePoint Designer if the Web server hosting the Web site supports one of these remote-authoring technologies:

- FTP
- WebDAV
- FPSE
- SharePoint

Depending on the remote-authoring technology that your Web server supports, some of the features of SharePoint Designer are disabled. SharePoint Designer offers its full capability for SharePoint sites. The SharePoint Designer features that rely on SharePoint aren't available for FPSE sites. Similarly, the features and components that rely on FPSE aren't available on FTP or WebDAV sites.

After you open a Web site in SharePoint Designer, you can use the Site Settings dialog box to determine the type of Web server that hosts the Web site. As with many other dialog boxes in SharePoint Designer, many of the options in this dialog box might be disabled depending on the Web site that you open. To open the Site Settings dialog box, as shown in Figure 1.4, choose Site ⇨ Site Settings.

Within the Site Settings dialog box are several tabs:

- **General:** This tab is used to display the version information about the Web server and version information about the FPSE or SharePoint installed on the Web server. You can use this tab to disable the use of hidden metadata for non-SharePoint and non-FPSE sites. Also, you can enable lightweight document check-in and check-out using this tab.
- **Preview:** You can use this tab to specify if you want to use the Microsoft ASP.NET Development Server to preview ASP.NET Web pages on your machine. SharePoint Designer installs the Microsoft ASP.NET Development Server during its setup. This option is useful if you don't have a Web server machine (ASP.NET pages need a Web server for processing) and would like to see a browser-based preview of the ASP.NET pages you create with SharePoint Designer.

CROSS-REF For more on ASP.NET, see Chapter 10.

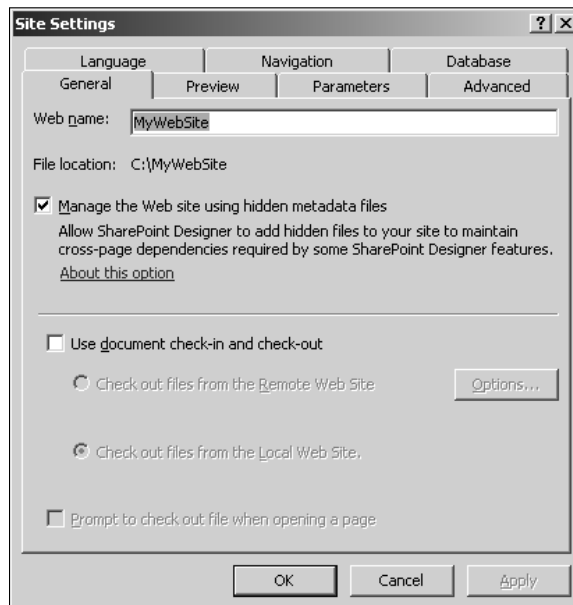
- **Parameters:** Allows you to create FPSE-based parameters for your Web site. These parameters can be used later by FrontPage components, such as substitution components. This tab isn't useful for Web sites that don't allow hidden metadata.
- **Advanced:** Allows you to set the default validation script language for FrontPage validation components. Also, it provides a check box to show hidden files and folders (certain metadata files) inside the Web site. Also, this tab isn't useful for Web sites that have hidden metadata disabled.
- **Language:** Allows you to set the default page encoding for the Web site. For Web sites with hidden metadata, this tab allows you to set the language of the server messages.
- **Navigation:** Allows you to customize the text labels on FrontPage link bar components. This tab isn't available for Web sites with disabled hidden metadata.

CROSS-REF For more on FrontPage components, see Chapter 9.

- **Database:** This tab is for backward-compatibility with Web sites that have Database Results Wizard pages in them.

FIGURE 1.4

The Site Settings dialog box



Maintaining Web Sites

After you create a Web site, you likely want to manage it properly so that you don't lose the work you've completed and don't have to re-create anything. However, Web site maintenance isn't just about backing up and publishing content. You also have to decide which content to publish, which Web pages to keep, and which version of files to delete.

SharePoint Designer offers a number of options for Web site maintenance. Depending on the type of Web site that you're working on, you have a set of tools available in SharePoint Designer to perform housekeeping and Web site management.

Understanding basic site management for non-SharePoint sites

The first operation that you need to perform when your Web site is ready is to publish it to a remote Web server by using an Internet address (URL) that your hosting provider or Web site administrator grants for your Web site. Publishing is the process of making the Web site content available for viewing by others. SharePoint Designer allows you to publish content from one Web server to another, from a disk-based location on the local computer to a Web server, or from a Web server-based location to a folder on your local computer (called reverse publishing).

CROSS-REF For more on managing non-SharePoint sites, see Chapter 23.

Publishing

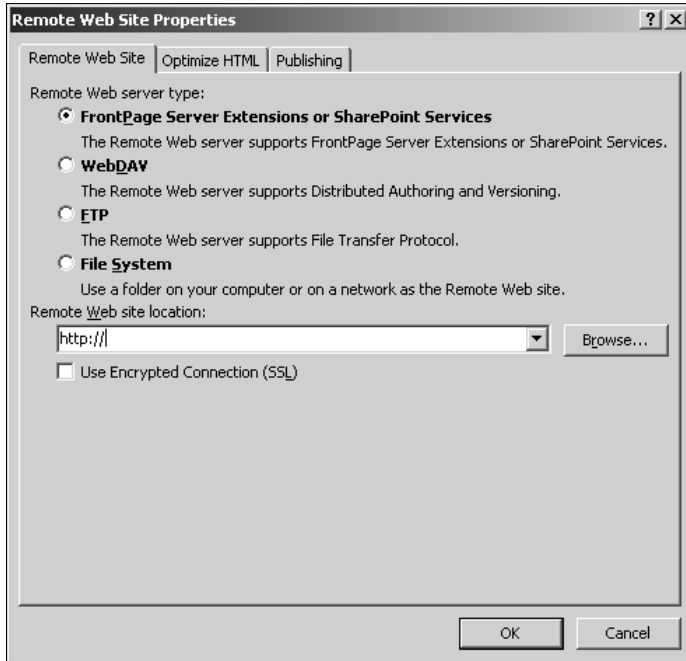
SharePoint Designer offers an easy-to-use, intuitive mechanism for publishing content from one location to another. You can access the SharePoint Designer publishing interface by choosing File → Publish Site. The Remote Web Site Properties dialog box, as shown in Figure 1.5, opens in the foreground, and the Remote Publishing pane opens in the background of the SharePoint Designer environment.

The Web site that you have open in SharePoint Designer becomes the local Web site for publishing and is displayed in the Local Web Site section of the Remote Publishing pane. You then use the Remote Web Site Properties dialog box to specify the type of remote Web server, the location of the remote Web site, and the settings for the publishing operation.

After the remote Web site is set up, you can either publish from local to remote, publish from remote to local (reverse publishing), or synchronize the content of the two Web sites.

FIGURE 1.5

The Remote Web Site Properties dialog box



Import and export content

SharePoint Designer provides two mechanisms for importing and exporting content, which are available by choosing File ⇄ Import or File ⇄ Export:

- **Import Site Wizard:** Use this wizard to import content from a variety of locations. Choosing to use FPSE, WebDAV, or FTP as an import location in this wizard links you back to the Remote Publishing pane for SharePoint Designer. However, the HTTP option allows you to download Web pages and content from any HTTP location on the Internet.
- **Personal Web Packages:** This allows you to package a Web site into a compressed file (.fwp) that can then be exported into another site. While exporting and importing contents of a Web site, SharePoint Designer allows you to choose the content you want to compress in the package for exporting and then choose content you want to import.

Versioning

Depending on the nature of a Web site, the SharePoint Designer interface allows a designer to check-out a document so that it becomes locked — preventing others from working on the Web site — modify and update the document, and save and then check-in the document so that the updated document is available for other users.

Versioning and document check-in and check-out features in SharePoint Designer mostly rely on the Web server configuration and type. Although basic check-in and check-out are available for disk-based (using hidden metadata), FTP, and WebDAV sites, the advanced versioning features are provided by Web sites that use FPSE or SharePoint. For example, you can enable basic document check-in and check-out (if you have hidden metadata allowed) by clicking the Use document check-in and check-out check box in the General tab of the Site Settings dialog box.

Reporting

You use the Reports pane in SharePoint Designer to access many of the reports that it generates for Web sites. Choose Site ⇨ Reports to open the Reports pane. Besides site and usage reporting, SharePoint Designer also offers these reporting features:

CROSS-REF For more on the Reports pane, see Chapter 25.

- **CSS reports:** You can have SharePoint Designer generate reports that show details about unused styles, undefined classes, mismatched cases, and CSS usage while implementing CSS. These reports help in consolidating the style sheets being used inside the Web site more efficiently.
- **Accessibility reports:** Using the Accessibility Checker in SharePoint Designer, you can find out if your Web pages are compatible with the accessibility Web standards. You can use the report generated to correct issues and improve accessibility.
- **Compatibility reports:** You can use these reports to find out if your Web pages are compatible with the Web browser and CSS schema you're targeting your Web site for.

Authoring SharePoint sites

With SharePoint sites, although most site management operations are performed using the SharePoint administration Web sites and tools, limited backup and restore and import and export capabilities are offered by SharePoint Designer.

In the user interface, SharePoint Designer treats SharePoint sites almost like any other Web site. So, you can perform most authoring and site management operations on SharePoint sites in manners similar to non-SharePoint sites. However, SharePoint Designer provides a lot of new features in collaboration with SharePoint that make the authoring and management experience a lot more different and enriching.

Here's a brief introduction to some SharePoint-specific features of SharePoint Designer:

- **SharePoint content:** SharePoint Designer allows the creation of almost all SharePoint content, including sites and subsites, page layouts and master pages, lists and document libraries, and surveys. It also offers the Document Library View Page Wizard and List View Page Wizard that assist in creating view pages for SharePoint document libraries and lists.
- **Workflows:** SharePoint Designer has an advanced Workflow Designer that can be used to create, compile, and deploy declarative workflows for SharePoint sites.
- **Data forms:** With SharePoint sites, SharePoint Designer enables a complete menu called Data View that offers capabilities to work with data sources provide by SharePoint. It offers the Data Form Web Part that can be used to create views and forms for SharePoint data sources.
- **Web parts and SharePoint controls:** SharePoint Designer allows for inserting and modifying properties of SharePoint Web parts and Web part zones. Also, it exposes a number of SharePoint controls as well as page and content fields that can be used on SharePoint Web pages.
- **Contributor settings:** These settings allow SharePoint administrators to enable or disable certain features and menu options in SharePoint Designer based on the permissions that a user has on a SharePoint Web site.
- **Back up and restore Web sites:** The publishing feature of SharePoint Designer isn't useable with SharePoint sites. Instead, you can use SharePoint Designer to package, back up, and restore SharePoint Web sites.

CROSS-REF For more on authoring and site management, see Chapters 5, 12, and 21.

Summary

In this chapter, you learned about the origins and birth of SharePoint Designer. You also learned about how to use SharePoint Designer to create basic Web pages and manage settings for Web sites. You learned how SharePoint Designer creates and manages hidden metadata for Web sites and whether to use raw Webs. You were also exposed to the remote-authoring abilities of SharePoint Designer as well as SharePoint and non-SharePoint authoring and site management features.

