Microsoft FrontPage to Expression Web

Transitioning to Microsoft® Expression® Web from Microsoft Office FrontPage® 2003

Published: March 2007

For the latest information, please see http://www.microsoft.com/expression

Microsoft

Microsoft Expression

Summary

FrontPage is being replaced by Microsoft Expression Web, a professional Web authoring tool with features that help you design, develop, and maintain exceptional standards-based Web sites. Expression Web's interface is similar to FrontPage, making it easier for FrontPage users to transition to Expression Web, but Expression Web comes with many new, different tools—and the ability to customize the interface layout—to help Web designers and developers work more efficiently.

FrontPage users who are interested in bringing their Web sites to higher levels of sophistication and standards compliancy will find Expression Web an invaluable tool for developing Web sites.

Contents

Introduction

After ten years of being an award-winning Web authoring tool, FrontPage has been discontinued. FrontPage has been among the most popular what-you-see-is-what-you-get (WYSIWYG) Web site creation tools since Microsoft released FrontPage 97 in late 1996. With its innovative use of server-side script and familiar user interface, FrontPage enabled users to create Web sites quickly and easily.

As the Internet industry has grown, so has the level of sophistication of the average Web site, and so have the expectations of Web designers and developers regarding the tools at their disposal. More and more, the industry is moving toward standards-based Web design as defined by the World Wide Web Consortium (W3C). FrontPage served a significant need during its product lifetime. However, as the Web has evolved, so have the needs and expectations of Web designers and developers regarding tools such as FrontPage.

FrontPage is being replaced with two great new tools for application building and Web authoring. For FrontPage users who work with sites built on the Microsoft SharePoint® platform, there is Microsoft Office SharePoint Designer 2007. (For more information about SharePoint Designer, see the SharePoint Designer home page on Office Online.) For FrontPage users who develop non-SharePoint sites, there is Microsoft Expression Web, a professional Web authoring tool with features that help you design, develop, and maintain exceptional standards-based Web sites.

With Expression Web and FrontPage installed on the same computer, you can continue to manage your FrontPage Web site and use the new tools and features included in Expression Web to take your FrontPage to the next level by transitioning to standards-based practices.

Expression Web isn't simply FrontPage in a fancy new package with a new name. Built with Web standards in mind, there are significant differences in how Expression Web works, all of which relate to building sites that are up-to-date with today's standards and technologies.

To help you make the leap not only to Expression Web but also into the world of Web standards, this document:

- Outlines the differences between FrontPage 2003 and Expression Web.
- Introduces the Expression Web workspace.
- Discusses Web standards.
- Provides information on how to work with your existing site, including FrontPage Web components, themes, and shared borders, in Expression Web.
- Introduces ASP.NET support and data integration available in Expression Web.

In an effort to provide you with practical, useful information, we've included links to third-party sites and solutions. **Please note:** Microsoft provides this third-party information to help you find the technical information that you need. Microsoft does not guarantee the accuracy of this third-party information. Third-party Web addresses (URLs) are subject to change without notice.

Differences between FrontPage and Expression Web

Expression Web builds on FrontPage 2003 technologies to provide an unprecedented level of support for creating standards-based Web sites. However, one of the objectives of Expression Web is to help you create high-quality Web sites based on today's standards and technologies. This means that non-standards-compliant features and functionality that were familiar tools in FrontPage do not exist in Expression Web.

Web Components

One of the first things you might notice as a FrontPage user is the absence of Web components in Expression Web. While Web components were an important feature in early versions of FrontPage, they were also code heavy (resulting in slow download times), difficult to customize, and they generated proprietary code that was far from standards-compliant. Much of the functionality that FrontPage Web components provided are now provided in new and often easily customized ways by Web-based services. For example, there are now multiple sites that make it possible to store, sort, and share your photographs in ways that were not possible when the Photo Gallery component in FrontPage was developed.

If your existing site uses Web components, you can still edit those components using Expression Web. However, you won't be able to add new Web components. For more information on working with existing Web components in Expression Web, or to find new and better ways to provide similar functionality, see the section entitled Working with FrontPage Web components with Expression Web.

FrontPage Server Extensions

FrontPage Web components relied on FrontPage Server Extensions, a set of proprietary scripts that were installed on the server. FrontPage 2003 had already started the move away from FrontPage Server Extensions, supporting only those Web components that were compatible with the FrontPage 2002 Server Extensions.

Expression Web does not require any server-side technology such as FrontPage Server Extensions. However, you can use Expression Web to manage and even modify your FrontPage Web site, including those sites that reside on servers that have FrontPage Server Extensions installed. As noted earlier, you will no longer be able to add new Web components that rely on FrontPage Server Extensions using Expression Web. For more information on working with existing Web components in Expression Web, or to find new and better ways to provide similar functionality, see the section entitled Working with FrontPage Web components with Expression Web.

Note: For more information on how long FrontPage support will be available from Microsoft, see the lifecycle information page for Office products and select your version of FrontPage. For FrontPage Server Extensions, choose the version of FrontPage that coincides with the version of the FrontPage Server Extensions that are installed on your server.

Themes and Shared Borders

Themes and shared borders offered quick and easy ways to style pages and keep the look and feel consistent throughout a Web site. However, they created proprietary code that was not standards-compliant and, having been held over from earlier versions of FrontPage, were woefully out-of date. In fact, both themes and shared borders were being phased out of FrontPage 2003. Neither themes nor shared borders exist in Expression Web, but you can still edit existing pages with FrontPage themes or shared borders applied. With new cascading style sheet (CSS) tools, Dynamic Web Templates, and ASP.NET master pages, you'll also find that Expression Web has a number of options that offer much of the

functionality that themes and shared borders provided, but with more power, more control, and the ability to create standards-compliant code. For more information on managing your theme-based and shared border-based FrontPage sites in Expression Web, see the section entitled Themes and Shared Borders.

Database Interface Wizard

The FrontPage Database Interface Wizard requires FrontPage Server Extensions (FPSE) and is not available in Expression Web. However, Expression Web supports ASP.NET 2.0, including drag and drop ASP.NET controls that help you connect to and display content from your database. For more information, see the section entitled Database Connections.

Introducing Expression Web

Microsoft Expression Web is a new Web design and development application providing powerful tools for producing high-quality, standards-based Web sites. Expression Web has strong CSS support and takes full advantage of the power of ASP.NET 2.0 with integrated support for server and user controls.

Expression Web is designed for creative professionals. Experienced WYSIWYG Web designers will also appreciate the extensive features of Expression Web. No technical background is required to start using Expression Web. Convenient task panes and menus help users build attractive and compelling Web pages without knowing any markup or server code.

Standards-based Web sites

When you use Expression Web to create and manage your Web site, you have a variety of tools at your disposal to help ensure that your Web site conforms to the latest W3C specifications. For example, once you have chosen a target standard for your Web site, real-time standards validation alerts you to errors that break your chosen standard. For more information, see the section entitled Building Standards-Based Web Sites with Expression Web.

Sophisticated CSS tools and support

Expression Web supports the creations of CSS-based Web sites through the use of sophisticated CSS tools. In addition, the Expression Web rendering engine provides a preview of your CSS-formatted site in Design view. The CSS tools in Expression Web can help you format your Web sites by using CSS, even if you have little or no prior knowledge of CSS. For more information on how to use the CSS tools in Expression Web, see the Appendix A: Using Expression Web to Create Styles.

ASP.NET support

Expression Web is the first WYSIWYG Web design tool to support the ASP.NET 2.0 release. If your Web host or ISP supports ASP.NET 2.0, you can take advantage of the power of ASP.NET 2.0 by using tools such as ASP.NET master pages, drag and drop ASP.NET controls, no-code data binding, and the ASP.NET Development Server, which allows you to preview your ASP.NET site directly on your computer. This built-in support for ASP.NET makes Expression Web a powerful choice for building Web sites on the .NET platform.

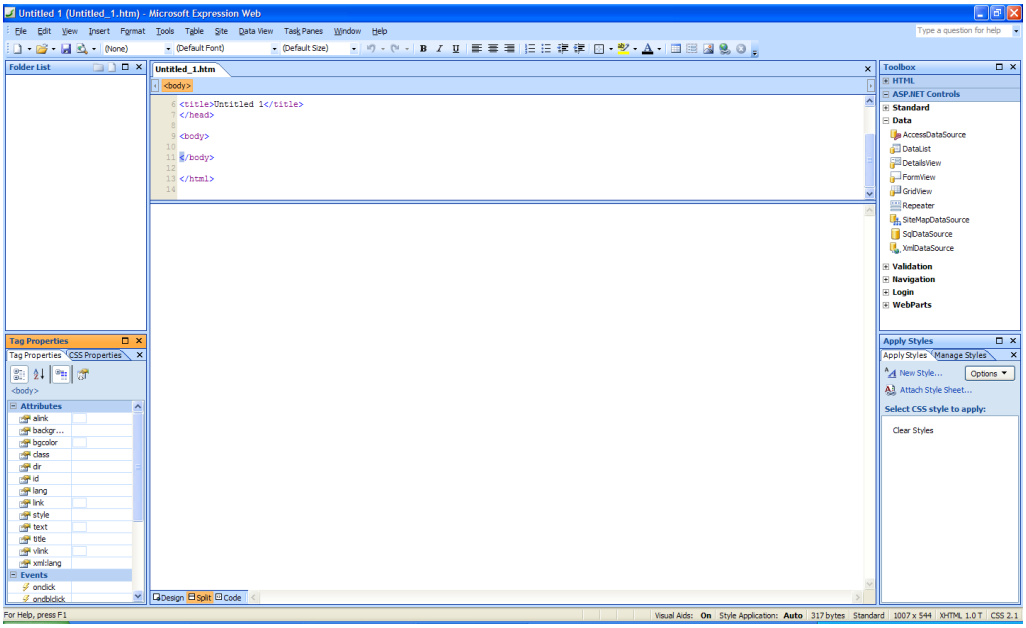
The Expression Web Interface

If you are familiar with FrontPage, then you will find Expression Web easy to use. Like FrontPage, Expression Web includes a Folder List, the Web Site tab, Design, Code, and Split views, as well as many of the same toolbars where they were located in FrontPage 2003. However, Expression Web also has many new and powerful features that take Expression Web to the next level of Web design. To help you become more familiar with Expression Web, the following sections introduce three particular features:

- Task panes
- Visual aids
- CSS tools

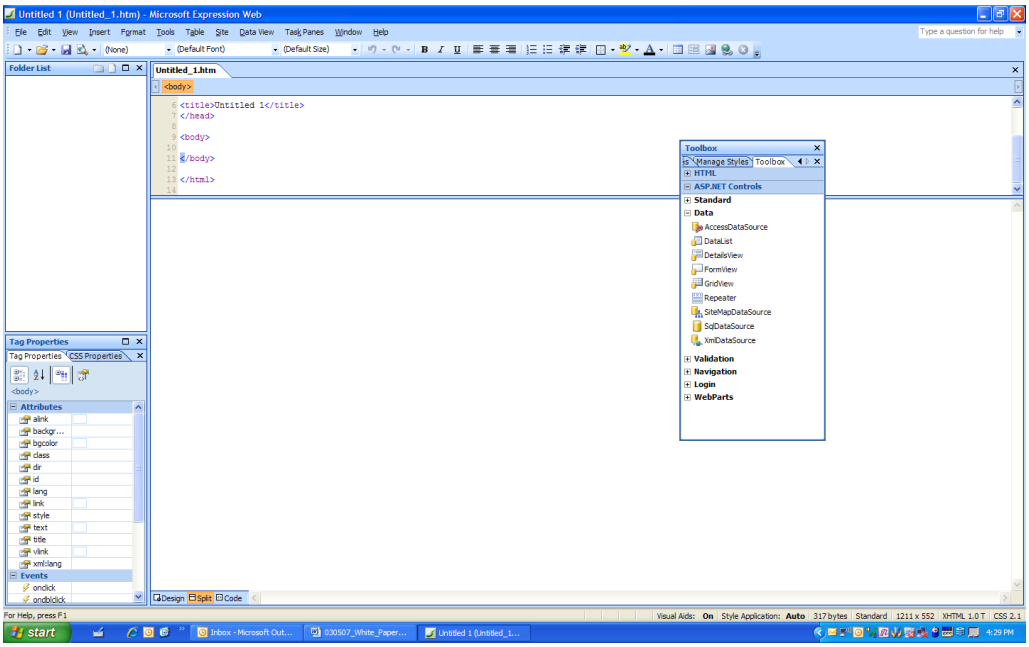
Task Panes

FrontPage 2003 had a single task pane that displayed a variety of tools, including the clipboard, themes, new file options, and more, depending on your context. With Expression Web you can have multiple task panes open at a single time and you can change the location and size of each task pane to suit your needs. When you arrange the task panes in a layout that works for you, Expression Web automatically uses this layout the next time you start the program.

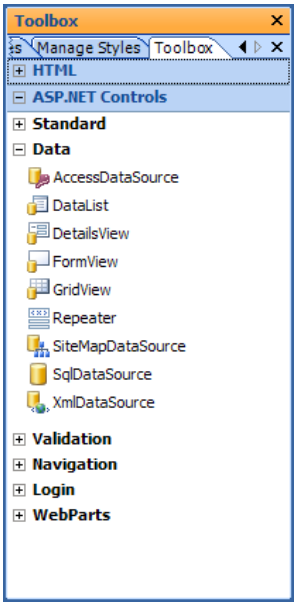


By default, task panes are on the left and the right, with the Design surface in the center.

In addition, task panes can be docked in the left column, right column, or below the editing pane. They can also float independently on top of your screen, or stack together with other task panes.



Task panes can be set to float anywhere in the window.



Task panes can also be stacked. For example, the Manage Styles task pane and the Toolbox task pane in this illustration are stacked.

You can find every task pane on the **Task Panes** menu. Some of the new task panes available in Expression Web include:

- **CSS Properties task pane:** Allows you to view and edit any CSS property that has been set for a particular element on a page.
- **Tag Properties task pane:** Allows you to view and edit any attributes of a particular element.
- **Apply Styles task pane:** Shows all of the available styles on a page's style sheets and allows you to apply them to an element with one click.

- **Manage Styles task pane:** Allows you to organize, move, rename, and otherwise manage styles defined in a style sheet.
- **Accessibility task pane:** Shows results of running the Accessibility Checker.

Visual Aids

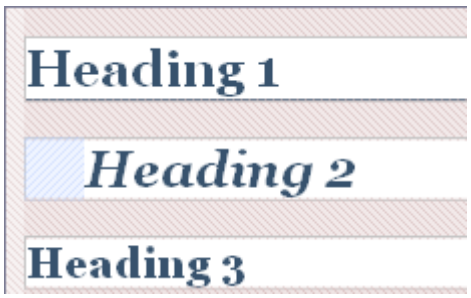
Visual aids (on the **View** menu, click **Visual Aids**, and then click **Show**) help you see empty or invisible elements and elements with hidden borders. You can also use visual aids to find elements that have hidden or invisible styles, as well as ASP.NET controls that aren't visible on a page. Finally, you can use visual aids to see which tags are used around specific content, and you can see the size of margins and padding around your tags.

- **Block Selection:** When this option is turned on, clicking within block-level elements on a page (such as paragraphs, headings, and list items) will show a dotted rectangle around the element and a tab displaying the name of the tag and any classes or IDs associated with the tag. Clicking the tab will allow you to select the entire element as well as visually see the margins and padding. You can drag the corner and side handles to resize the element or drag the margin and padding handles to resize those, as well. The tab also allows you to reposition absolutely-positioned elements.

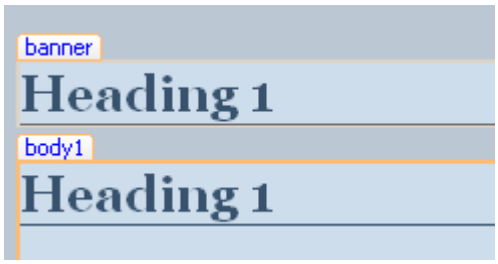
- **Visible Borders:** When this option is turned on, a faint dotted line appears around elements (such as table elements) that have hidden borders.

- **Empty Containers:** When this option is turned on, a dotted line appears around empty elements instead of collapsing them by default.

- **Margins and Padding**: When this option is turned on, pink shading appears for margins and blue shading appears for padding.



- **CSS Display:none Elements:** When this option is turned on, you'll be able to see the elements that have been hidden by the CSS **display:none** property.
- **CSS Visibility:hidden Elements:** When this option is turned on, you'll be able to see the elements that have been hidden by the CSS **visibility:hidden** property.
- **ASP.NET Non-Visual Controls:** When this option is turned on, Expression Web shows a rectangle for ASP.NET controls which otherwise don't display in Design view.
- **ASP.NET Control Errors:** When this option is turned on, error messages will display for ASP.NET controls.
- **Template Region Labels:** If you use Dynamic Web Templates or ASP.NET master pages, this option shows a border around editable regions and a tab with the name of the region.



CSS Tools

Expression Web provides you with a comprehensive set of tools to create, apply, and manage styles and cascading style sheets. These tools include:

- **New Style and Modify Style:** Design a new or existing style and preview the style's appearance as you design it.
- **Apply Styles:** Create, modify, apply, remove, or delete styles, and attach or remove an external CSS.
- **Manage Styles:** Move a style from an internal CSS to an external CSS, or vice versa.
- **CSS Properties:** See the styles that are used by the current selection in your Web page in order of precedence.
- **Style toolbar:** Apply or remove class-based or ID-based styles, or create and apply new undefined styles.
- **CSS Reports:** Generate a report of either CSS errors or CSS usage on one or more pages or within an entire site.
- **Style Application toolbar:** Use the toolbar when authoring CSS in Manual style application mode.
- **IntelliSense for CSS:** Increase your productivity when authoring and applying CSS in Code view.

Web Standards and Expression Web

Web standards refer to a list of standards or recommendations established by the W3C for creating and publishing Web sites. These standards are generally agreed to be best practices for Web site development and take into consideration such issues as accessibility for people with disabilities, cross-browser compatibility, and platform compatibility. When Web sites conform to W3C standards, they are more likely to be accessible to a greater number of people using diverse devices, browsers, and platforms. Expression Web features state-of-the-art tools for ensuring that your Web sites are compliant with these standards.

As accessibility and Web standards have become more important, the way Web sites are built has changed as well. How your content is marked up—the HTML—is as important as the design (the presentation or display) of the site. Part of the current recommendations for XHTML provides guidelines for how HTML code is written and structured, and for using CSS, which separates the presentation from the HTML. When you separate content from presentation, it is much easier to create Web sites that are accessible not only to people with disabilities by adhering to the Web Content Accessibility Guidelines (WCAG), but also to other form factors, including printers, handheld devices such as cell phones and PDAs, and across different browsers and browser versions. With a standards-compliant site and multiple style sheets, you can more easily provide alternative designs for other form factors instead of having to try to maintain several different versions of your site.

DOCTYPEs

At its core, creating a standards-compliant site essentially means making sure that your code conforms to a specific set of rules identified by a specific DOCTYPE. The DOCTYPE declaration—a line of code at the top of the HTML document—declares which set of rules your HTML is designed to follow. There are DOCTYPES—rules—for HTML as well as XHTML, and for the strict as well as the transitional application of those guidelines. The following are examples of DOCTYPE declarations:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

DOCTYPE declaration for HTML 4.01 Transitional

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

DOCTYPE declaration for XHTML 1.0 Strict

For more information on DOCTYPE declarations and which one to choose, see Fix Your Site with the Right DOCTYPE.

Once you've decided which set of guidelines that you want to follow, you can then test, or "validate," your site to see how closely they follow the guidelines. A compliant Web site is one that passes the validation test. An accessible Web site is one that follows the accessibility guidelines. Expression Web has built-in validation tools, code error notifications, and more to help you create a Web site that follows Web standards.

Building Standards-Based Web Sites with Expression Web

This section covers best practices for creating a standards-compliant, accessible Web site when working with Expression Web.

Meaningful markup

Separating content from the presentation—in other words, separating the markup and text from the visual display—is a fundamental principle in building standards-compliant Web sites. Most Web professionals, however, take this idea one step further. For your markup to be meaningful and to be accessible across the largest range of devices, you'll want your content to be marked up *semantically*. Semantic markup is meaningful even if all styles and formatting are removed. For example, you might use an <h1> tag to mark up the title of a document because text marked up with the H1 element is a first-level heading. You could also mark up the title with a <p> tag, and then apply styles to make it display in a browser exactly like the H1 element would. However, if the style sheet is not applied, the text in the H1 element still looks like a title or heading, but the text in the P element looks just like a normal paragraph instead of a title, making the title less meaningful in comparison to the rest of the text.

This is true of the full range of HTML markup that is available to you. Use the different level heading tags available in HTML (H1, H2, H3, H4, H5, H6) as they were meant to be used—for creating headings and subheadings. If you use the proper tags throughout, you can format your Web content in a way that conveys meaning through its structure. Don't use different heading tags just to get different-sized text. If you want to emphasize text that isn't a heading or subheading, use the or tags and then create a class to apply a format to the text so the text appears the way that you want it to appear.[*]

There are several additional benefits to using semantic mark up:

- The HTML/XHTML standards advise you to use the H1 element for first-level headings. By using H1, you would be following a recognized standard, which means that browsers, text-to-speech readers, and other assistive devices would know how to deal with the heading. Your site would be more accessible.
- Your code would be more search-engine optimized. Search engines would be able to recognize your H1 element as a heading and would put more weight on the text in the heading.
- Your code is cleaner.

For best results, before you use a graphics editor or a style sheet, you'll want to mark up your content semantically and then make sure that your document follows the rules for your declared DOCTYPE. This will help you focus on creating a structured document by using the full range of HTML elements available for marking up text, including blockquotes, ordered and unordered lists, data tables, headings, subheadings, and more. After creating your content-only pages, you can then begin formatting the site by using style sheets or creating and then attaching ASP.NET master pages or Dynamic Web Templates.

---

[*] The and tags are semantically meaningful ways to emphasize text. Most browsers display strongly emphasized text in bold and emphasized text in italics. However, according to the W3C XHTML recommendation, they should be used instead of the italic (<i>) and bold (<b>) tags because they convey emphasis, rather than visual difference. This is especially important when browsing web sites using a text reader rather than a visual display. By default, Expression Web inserts the and tags in place of <b> or <i> tags when you use the toolbar or keyboard shortcuts to bold or italicize text.

For more information on semantic markup, see the article entitled Integrated Web Design: The Meaning of Semantics (Take I).

Style Sheets Instead of HTML for Formatting and Layout

Many Web developers share the common experience of going through two general stages of learning about CSS. The first stage is learning how to format text and images, including font families, colors, padding, background images, and borders. (For an introduction to using CSS to apply styles, see Appendix A: Using Expression Web to Create Styles.)

The next stage has a steeper learning curve, but is the direction that the Web is moving in: learning how to use CSS instead of tables for layout.

There are a lot of good reasons not to use tables for layout. One reason is that the table structure itself is semantic, meaning that tables are meant for displaying data, not for facilitating the layout of a page. To use a table semantically, tables should only be used to display data. In addition, tables used for layout, especially nested tables, are especially difficult to understand when a visitor is using an audio-based browser. Sites using tables for layout are inaccessible to audio-based browsers and other assistive devices, as well as to devices with displays that are smaller than the tables.

You may not be ready to immediately transform your table-based layouts into CSS based layouts. However, to create an accessible, standards-based web site, it's still a good idea to begin using CSS to apply styles to your web content. If you are using tables for layout, make them more accessible by using as few tables and table cells as possible, and make sure that your tables make sense if a text-to-speech reader were to read the contents of the table across each row and down the page. (For more information on how to make layout tables more accessible, see the Tables for layout section of the W3C's Web Content Accessibility Guidelines. For more information on how to begin using CSS to create table-free layouts, see CSS Layout Techniques for Fun and Profit.)

There are many resources available—books, online articles and tutorials, videos, and more—to help you learn CSS. Here are some resources that you might find helpful:

- Beginner's guide from a seasoned CSS designer
- Veerle's Interesting CSS Links
- Smashing Magazine's CSS Techniques, Tutorials, Layouts

Customized Displays for Multiple Media Types

One of the benefits of using CSS is that you can create different style sheets for different types of devices and then link to multiple style sheets that use the media attribute to identify those devices. Currently, the W3C recommendation includes the following list as recognized media descriptors:

- **Screen:** Intended for standard computer screens.
- **TTY:** Intended for media, such as teletypes, terminals, or portable devices with limited display capabilities.
- **TV:** Intended for television-type devices such as MSN TV.
- **Projection:** Intended for projectors.
- **Handheld:** Intended for handheld devices such as PDAs.
- **Print:** Intended for printed documents and for display using the Print Preview command.
- **Braille:** Intended for braille-based feedback devices.

- **Aural:** Intended for text readers/speech synthesizers.
- **All:** Suitable for all devices.

The following are examples of media attributes used in the CSS link in the HEAD of an HTML document:

```
<link rel="stylesheet" type="text/css" media="screen" href="screen.css" />
<link rel="stylesheet" type="text/css" media="print" href="print.css" />
<link rel="stylesheet" type="text/css" media="handheld" href="mobile.css" />
```

Using the above CSS links as an example, the screen-optimized style sheet (screen.css) contains all of the styles that are applied when the page is viewed on a computer screen. The print-optimized style sheet (print.css) might hide the navigation bars and banner ads and set the margins to fit on a print page, so the elements that work on a screen but clutter the printed page can be omitted. The handheld-optimized style sheet (mobile.css) might use a one-column format instead of a three-column format and use images optimized for a smaller screen.

You can then link to each of these style sheets in the same page just as you would link a standard style sheet, making sure that the media attribute is included in each sheet and that the correct style sheet for each media is referenced. The media attribute does the work of identifying what device is being used to access the site, and subsequently which style sheet to apply.

Checking for Compatibility and Accessibility

One of the many benefits of using Web standards to design your site is enhanced cross-browser compatibility and accessibility. Expression Web includes built-in tools to help you check for compatibility and accessibility errors.

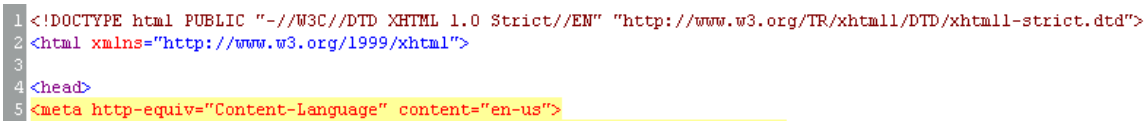
In order to use the compatibility and accessibility checkers, you should always specify a DOCTYPE for your Web page. A DOCTYPE, as discussed earlier, identifies the specific HTML standard that your document follows. If your page does not already have a DOCTYPE specified, you can go into the HTML code and insert the DOCTYPE manually. Expression Web comes with code snippets to allow you to do this easily:

1. In **Code** view, place the insertion point at the top of the page and press CTRL+ENTER.
2. Select the DOCTYPE declaration that you want to insert from the list.

Once you have the DOCTYPE specified, Expression Web will be able to better help you be aware of coding errors that don't meet the specifications of that DOCTYPE. In Code view, code errors are highlighted in yellow. For example, if a <meta> tag is missing the closing slash, Expression Web will highlight the tag in yellow to let you know that there is a problem with the code. If you hover the pointer over the highlighted code, Expression Web provides a screen tip explaining the error.

Similarly, if you hand code and have a typo in one of your HTML tags, Expression Web will put a red squiggly line underneath to let you know that there is a problem. Again, you can hover over the underlined content to view a screen tip telling you what the error is.

The red underline indicates HTML incompatibility.



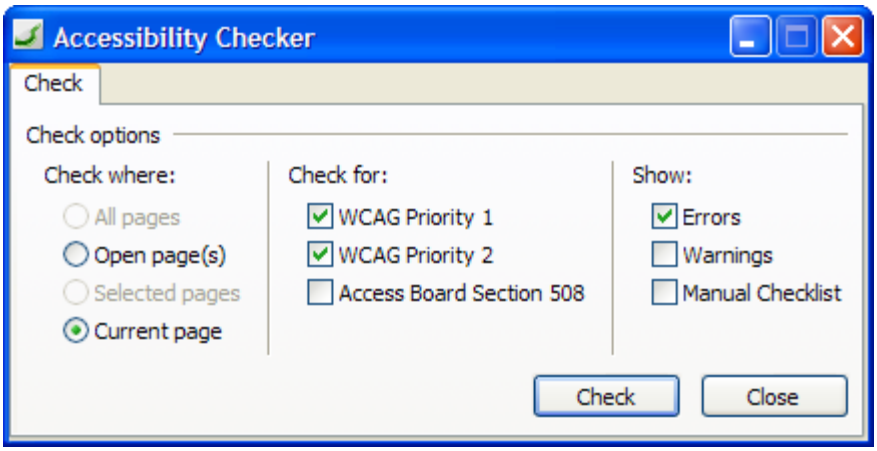
The META element without an end tag highlighted in Code view indicates a code error.

Expression Web also displays the HTML Incompatibility Detected icon and the Code Error Icon in the status bar if it detects a problem with the HTML code. This allows you to be notified of code errors while working in Design view.

The HTML Incompatibility Detected icon and the Code Error icon in the Status bar.

Expression Web also helps you to meet Web standard and accessibility guidelines in the Compatibility and Accessibility task panes. Both task panes run reports when you click the green arrow. You choose what specification you want to test the site against. When the report is complete, the errors are then listed, making it easier to locate and address each one.

For example, in the Accessibility Checker, you can select the range of pages that you want to check and then select the specific guidelines that you want to test your site against. You can then set the report to show errors (compliance errors), warnings (potential problems), or a manual checklist (issues that require human verification).



Similarly, the Compatibility checker allows you to select the range of pages to check and for which standards to test.

After publishing your Web site, you can check your site one more time by using some of the online tools available for validating Web sites:

- W3C HTML Validator
- W3C CSS Validator
- Accessibility compliance checker

Apart from testing your site against the various Web standards, you should also use Expression Web to test your Web site in multiple browsers. Install the various browsers you want to test your site against. To add them to your browser list, on the **File** menu, click **Preview in Browser,** and then click **Edit Browser List.** After adding the browsers, you can then open the Web page that you want to check, and then on the **File** menu, click **Preview in Browser.** After that, select the browser and resolution that you wish to test your site against.

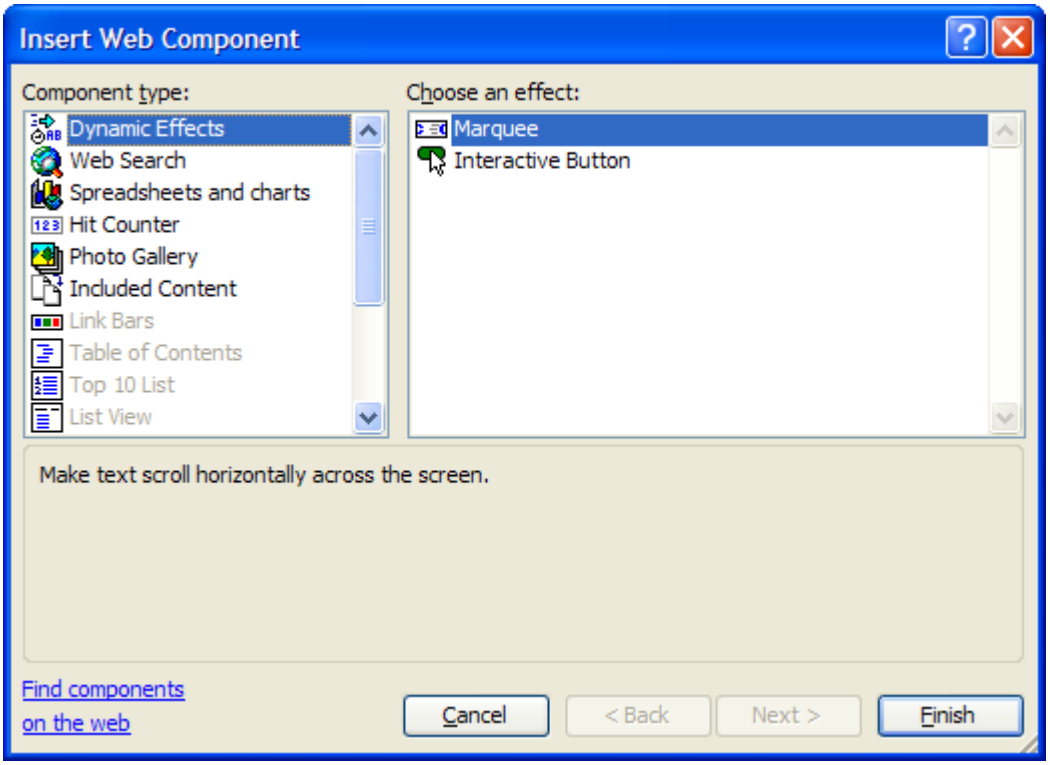
Working with FrontPage Web Components in Expression Web

FrontPage included many Web components, also called webbots, that enabled a lot of common functionality which beginning Web designers saw on other sites and wanted to add to their own sites. As mentioned earlier, Web components don't exist in Expression Web. Web components were notorious for not being standards–compliant and very difficult to customize. For FrontPage users transitioning to Expression Web, the good news is that Web components that require FrontPage Server Extensions (FPSE) will continue to work as long as the site remains on a server with FPSEs. In addition, almost all of the components can be edited in Expression Web by double-clicking the component in Design view.

Even though Expression Web supports your pre-existing Web components, you might want to consider moving away from FrontPage Web components for several reasons. First, they produce invalid code; second, your ability to customize Web components is severely limited; third, FrontPage Server Extensions have a limited life span; and fourth, you can't add new components using Expression Web. This section explains how to edit FrontPage Web components you may have in your site and also provides resources and suggestions to get functionality similar to that provided by unsupported FrontPage Web components.

The Insert Web Component Dialog Box

In FrontPage, you inserted Web components by clicking the **Web Components** command on the **Insert** menu to display the **Insert Web Component** dialog box.



The following paragraphs address each of the components as they appear in the **Insert Web Component** dialog box.

Dynamic Effects

The Dynamic Effects component type included Marquee and Interactive Button.

Marquee

The Marquee component simply added the <marquee> tag by using a dialog box that let you insert text and define attributes. The MARQUEE element was introduced by Microsoft as an extension to the HTML 3.2 specification, and is not included in any of the W3C's HTML or XHTML recommendations (meaning that it is not standards compliant). You can use still use the <marquee> tag, but your page won't be valid, and the text may not scroll as expected in browsers other than Internet Explorer.

There are also ways to create a similar scrolling effect by using a combination of HTML and JavaScript. Use your favorite search engine and use a key phrase such as "scrolling text." You will find that there are many free JavaScript sites that not only give you the script, but also give you instructions on how to implement the script in your site.

Interactive Button

Expression Web includes Interactive Button functionality. (On the **Insert** menu, click **Interactive Button.**) As with FrontPage, the Expression Web implementation of interactive buttons relies on images and JavaScript to implement the rollover effect. You can also create the same effect by using CSS, eliminating the need for additional images and script. For more information on how to use CSS to create your own interactive "buttons," see Appendix B: Using CSS to Style Text Links, and Appendix C: Formatting a bulleted list of links with CSS.

Web Search

The Web Search component type included Current Web and Full-Text Search.

Current Web

The Current Web Web search component allowed you to create a search form that then allowed users to search your entire site using keywords. In addition to relying on FPSEs, this component also required special implementation on the part of your server administrator. Many search engines now offer the ability to add "search this site" functionality to your current Web site. You can use your favorite search engine and use a key phrase such as "add site search" or "build your own search engine."

To add Live.com site search functionality to your Web site, see Add Windows Live Search to your Web site. To add Google site search functionality to your Web site, see Google Custom Search Engine.

Full-Text Search

Full-Text Search Web component required Windows® SharePoint Services. For more information on Microsoft Windows SharePoint Services, see Windows SharePoint Services or Microsoft Office SharePoint Server 2007.

Spreadsheets and Charts

The Spreadsheets and charts option embedded a Microsoft Office component into a Web site to display Microsoft Excel® spreadsheets or Office charts. Expression Web is not a part of the Office family of products. If you want to use an Office product to build Office-based information solutions, see if SharePoint Designer and either SharePoint Server or Windows SharePoint Services would better suit your needs.

Hit Counter

The FrontPage Hit Counter Web component inserted a component that tallied and displayed the number of times the page was visited. If you have a FrontPage Hit Counter component in

your site, you can still modify it by using Expression Web and double-clicking on the component. However, you cannot add a new hit counter component to a page.

It is worth noting that all standard hit counters, not just the FrontPage hit counters, are notoriously unreliable when it comes to getting accurate information about the number of visits to your site. The solutions below are more likely to provide you with more accurate information about traffic to your Web site.

Many hosting providers will provide some sort of server statistics tracking service. Contact your hosting provider to see if they have something that will allow you to keep track of page views and Web site use over time.

If your hosting provider doesn't have anything that allows you to track statistics, there are many other third-party providers that allow you to insert a small amount of code on your pages that then allow you to track stats through their service. You can use your favorite search engine and use a key phrase such as "hit counter" or "site statistics." Some examples include:

- Statcounter
- Shinystat
- Tinycounter

Photo Gallery

The FrontPage Photo Gallery component provided an easy way to insert a photo gallery with clickable thumbnails into a page. Existing photo galleries can be modified in Expression Web by double-clicking on the component.

The Photo Gallery Web component, in addition to being dependent on FrontPage Server Extensions, also generated invalid markup, and it was difficult to customize the display beyond a few built-in layout options.

Instead of using the Photo Gallery Web component, you can create your own photo gallery by optimizing the photographs for the Web yourself and creating a custom layout in Expression Web. You can also download a trial version of the soon-to-be-released Expression Media and create your own photo gallery, or you can select from the other popular photo gallery services and solutions, including but not limited to:

- Flickr: A free online photo gallery service. You can store your photos online and add information about each photo, which you can then display on your site.
- Gallery: A free, open–source application that allows you to manage digital images and publish them to the Web.

You can also use your favorite search engine and use a key phrase such as "online photo album" or "Web site photo gallery."

Included Content

The Included Content Web component enabled the inclusion of specific content based on certain conditions. These included:

- **Substitution** Inserted a component that showed the author, description, last changed date, or URL.
- **Page** Inserted a component that displayed the contents of one page on another page.
- **Page Based on Schedule** Inserted a component that displayed the contents of another page on the active page at specified times.

- **Picture Based on Schedule** Inserted a component that displayed a picture on the active page at specified times.
- **Page Banner Inserted** a page banner that was displayed as a graphical title based on the page theme or as text.

Replacements for the Substitution, Page Based on Schedule, and Picture Based on Schedule components require scripting or server-side programming.

Page Included Content

The Page Included Content Web component is a design-time component that made it easy to include the same content on multiple pages. As a design-time component, the component does not rely on FrontPage Server Extensions.

If you have existing Include Pages, double-click the component to access the **Include Page Properties** dialog box. Since the FrontPage code is not dependent on FrontPage Server Extensions, you can also use the same code to create new include pages in Expression Web.

You can easily create an Include Page code snippet for easy reuse later on.

To create the code snippet:

1. On the **Tools** menu, click **Page Editor Options.**
2. In the **Page Editor Options** dialog box, on the **Code Snippets** tab, click **Add**.
3. In the **Add Code Snippet** dialog box, in the **Keyword** box, type a unique keyword for the code snippet (**Include Page**, for example), which you can subsequently use to select the code snippet in the list of code snippets.
4. In the **Description** box, type a description to identify the code snippet. In the **Text** box, type or copy and paste the following code:

```
<!--webbot bot="Include" U-Include="filenamehere.htm" TAG="BODY" -->
```

To insert the Include Page code snippet:

1. In **Code** view, place the insertion point where you want the included page to appear, and then press CTRL+ENTER.
2. Select the **Include Page** keyword from the list of code snippet keywords, and then press ENTER.
3. Double-click the Include Page placeholder. In the **Include Page Properties** dialog box, click **Browse**, and then select the page that you want to include in the current page. Click **OK**.

Once you've inserted the Include Page code snippet into your page, you can edit it by double-clicking the component in Design view, locating the page that you want to include in the **Include Page Properties** dialog box, and then clicking **OK**.

Page Banner Included Content

The Page Banner is a component that displays the page title, either as text or within a theme-based banner graphic. In FrontPage, the title can be changed in Navigation view or by double-clicking on the page banner. In Expression Web, existing page banners are editable. However, new pages will not display the title because the pages must be added to Navigation view before displaying the page banner text.

Instead of using the Page Banner component, simply type your page title directly into the page. For semantic markup, you will want to use the H1 element. You can then apply a style to the H1 element by using CSS. By defining a background image by using CSS, you can create a title that resembles a FrontPage page banner, but one that is more standards-compliant and completely customized!

In the following steps, you are going to create a new single page Web site, then create a new HTML page, copy and paste an image into the Web site, and then create a new CSS page.

Note: You will be using this practice Web site throughout the rest of the examples in this paper.

To create the HTML page:

1. Create a new one page Web site by clicking **File** and then clicking **New**. In the **New** dialog box, click the **Web Site** tab, then click **General**, and then click **One Page Web Site**.
2. Copy the following code by selecting it and then pressing CTRL+C.

```
<h1 id="pageBanner">Page Banner</h1>
```

3. In the **Folder List,** right-click Default.htm and then click **Open**. (If the **Folder List** isn't visible, on the **View** menu, click **Folder List,** or press ALT+F1.) In **Code** view, place your insertion point where you want your page banner to appear and then press CTRL+V.
4. Save the file by clicking **File** and then clicking **Save**. In the **Save As** dialog box, in the **File name** box, type a new name for the file (such as PageBanner.htm), and then click **OK**.

To create the image:

1. Right-click the following image and then press CTRL+C.

2. Open your favorite image editing application, create a new file, and paste the image into the new file by pressing CTRL+V.
3. Save the image as pageBanner.jpg (JPG format).
4. Import the image into your new Web site by clicking the **File** menu, pointing to **Import**, and then clicking **File**. In the **Import** dialog box, click **Add File**. In the **Add File to Import List** dialog box, click the image that you just created, and then click **Open**. Click **OK**.
5. It is good practice to keep your images in a separate images folder. To create the folder, right-click in the **Folder List**, point to **New**, and then click **Folder**. Right-click the folder that you just created, and click **Rename**. Type **images**, and then press ENTER. Now click pageBanner.jpg, and drag it into the folder that you just created.

To create the CSS:

1. In the Web site that you just created, click **File**, point to **New**, and then click **CSS**.
2. Copy the following code by selecting it and then click CTRL+C.

```
/* the style below sets the page banner */
#pageBanner {
color: #336699;
text-transform: lowercase;
background: url('images/pageBanner.jpg') no-repeat;
padding: 20px 0px 20px 90px;
font: bold 36px Georgia, Times, serif;
}
```

3. Place the insertion point at the top of the new CSS page and paste the code into the page by pressing CTRL+V.

4. Save the page by clicking **File** and then clicking **Save**. In the **Save As** dialog box, in the **File name** box, type a new name for your CSS file (style.css, for example). Click **OK**.
5. To apply the style sheet to the Web page, with Default.htm open in **Design** view, click the style sheet in the **Folder List** and then drag the style sheet onto the Default.htm. The resulting page should look something like the following:

Note: The browser renders the text as all lowercase even though the text without the style applied is capitalized. This is because the value of the **text-transform** property is lowercase. Similarly, the image is applied to the text as a background image in the style sheet, so the image appears behind the text.

Now that you have created your page banner, you can insert it as an Include Page on the pages where you want it to appear and maintain it in the page that you just created, Pagebanner.htm. For more information on how to create Include Pages, see the section entitled Included Content.

Link Bars

In FrontPage, Link Bars provided a user-friendly way for you to maintain, manage, and edit navigation bars without having to modify every page. Link Bars provided graphical navigation buttons with easy-to-edit text. They were also used in conjunction with Navigation view to reorder and rename link bars, as well as to add new links.

The Link Bar Web component provided three types of link bars: a bar with custom links, a bar with next and back links, and a bar based on navigation structure. Each of these used FrontPage Server Extensions to maintain the navigation structure and/or sequence of pages.

You can still edit the properties of an existing FrontPage Link Bar—including style, page levels to display, and orientation—in Expression Web, but without Navigation view, you cannot create additional links or reorder existing links, nor can you automatically create navigation for a specific sequence of pages without creating those links within each page.

You can, however, easily create a custom link bar, save the link bar in a separate page, and then save it as an Include Page. That way you can maintain your navigation in a single page rather than having to maintain the links in every page in which the navigation is included.

The following will help you create your own navigation that you can use in place of a custom link bar.

Replacing a FrontPage Link Bar with your own links

You can easily create a list of links in Expression Web by using text links and CSS. In this example, you will create a link bar, and then create pages to link to by using the Insert Hyperlink dialog box. You may want to use your existing link bar as a guide for creating the following link bar.

1. Create a new page in the practice Web site that you created for the page banner by clicking **File**, and then clicking **New**. In the **New** dialog box, on the **Page** tab, click **General** and then click **HTML**. Click **OK**.
2. To create the list, do one of the following:

- For a horizontal list of links, copy the following code by selecting it and pressing CTRL+C. Note that each text item is separated with the pipeline character (|).

```
<p>Home | About Me | Photo Gallery | Contact Me</p>
```

- For a vertical list of links, copy the following code by selecting it and pressing CTRL+C. Although you can use paragraph breaks or line breaks between each link to create a vertical list, a bulleted list format is a more semantic and standards-friendly way to create Web site navigation.

```
<ul>
<li>Home</li>
<li>About Me</li>
<li>Photo Gallery</li>
<li>Favorite Links</li>
<li>Contact Me</li>
</ul>
```

3. Select each text item (Home, for example) and press CTRL+K to open the **Insert Hyperlink** dialog box. In the **Insert Hyperlink** dialog box, in the **Link to** column, click **Existing File or Web Page**. In the **Look in** column, click **Current Folder**. In the main window, click Default.htm. Repeat for each page.

   Note: All of the hyperlinks in the above example will link to Default.htm. If you want to create a new page for each link, in the **Link to** column, click **Create New document**. In the **Name of new document** box, type a name for your new document (for example, About.htm for the About me link). Under **When to edit**, check the **Edit the new document later** checkbox. Click **OK**. Repeat for each link.

For more information on how to style your links, see Appendix B: Using CSS to Style Text Links, and Appendix C, Formatting a bulleted list of links with CSS.

Table of Contents

The Table of Contents component created either a site map or a list of pages based on assigned categories. As with other FrontPage Web components, the two types of Table of Contents components can be edited in Expression Web. However, you cannot create a new Table of Contents component.

If you want a site map or table of contents on your site, you can create your own bulleted list or table of links, although you will need to update the page manually if you add pages to or remove pages from your Web site. You can also use a third party tool such as XML-Sitemaps, which generates XML or HTML-based site maps based on your file structure, and then you can include the generated page or XML data on your site.

Top 10 List

The Top 10 List components required FrontPage Server Extensions or SharePoint Services. You can create a similar list by reviewing your site analysis information. For more information on site analysis, contact your server administrator or Web host.

List View

The List View component required SharePoint Services. For more information, see SharePoint Designer and either SharePoint Server or Windows SharePoint Services.

Document Library View

The Document Library View component required SharePoint Services. For more information, see SharePoint Designer and either SharePoint Server or Windows SharePoint Services.

MSN Mapping

The MSN Mapping component embedded a map in your Web page. There are now many different services that provide you with similar functionality. You can also use your favorite

online mapping service to find a location, and then copy the URL to use as a link on your Web site. Use your favorite search engine and a key phrase such as “insert a map on a Web page” or “link to a map.”

MSN Components

MSN components inserted a search form or a stock quote in your Web page. To insert a search form in your Web page, see the section entitled Web Search. To insert a stock quote, use your favorite search engine and use a key phrase such as “stock quotes” or “insert a stock quote on a Web page.” Your favorite financial Web site may also have an RSS feed that you can include on your page. For more information, see the blog entry entitled RSS and Expression Web.

MSNBC Components

The MSNBC Web components displayed image links to news and financial sites from MSNBC, including business, living and travel, current events, sports, technology and weather. You can use Expression Web to insert an RSS feed for your favorite headline news. For more information, see the blog entry entitled RSS and Expression Web.

Additional components

The Additional components option is empty unless you have added custom components to a Web site you created with FrontPage. If you do have custom components, you should test your custom components to see if they will work with Expression Web. In addition, you should check that the component creator has produced a new, Expression Web version of the add-in that offers similar functionality.

Advanced Controls

The Advanced Controls option allowed you to embed objects such as Java applets, Adobe Flash movies, and other objects. For more information, see Information for Developers about Internet Explorer.

Themes and Shared Borders

FrontPage themes allowed users to obtain matching graphics, colors, and fonts for a site's link bars, page banners, bullets, and more. They provided an easy way to obtain rollover buttons without needing a graphics editor or JavaScript know-how, and created a consistent look across a Web site, while also allowing the user to change the look with a few clicks. Shared borders made it easy to have global content along the top, bottom, or sides of a Web page, keeping the user from having to modify every page. Newer Web technologies make it easy to have similar functionality without the invalid, clunky code and out-dated look and feel generated by themes or shared borders.

Instead of using themes, format your Web site with CSS. While there is a bit of a learning curve associated with CSS, you also gain much more flexibility and control over the design of your Web site than you have with FrontPage themes. Once you separate the content of the site (HTML) from the presentation (styles and formatting) by using CSS, you can then more easily change the look and feel of your entire site by modifying the attached style sheet.

Instead of using Shared Borders, use Dynamic Web Templates or ASP.NET master pages to create templates. These templates can contain the content common to all of the pages of your site such as copyright information, logos, headings, and menus, for consistency across all pages. Once you have created a template, editing the common content is as easy as editing a shared border: modify the template, and then update all of the attached pages to reflect the change. By combining Dynamic Web Templates or ASP.NET master pages with CSS-based design, you gain the flexibility and ease-of-use of themes and shared borders, but with more control, more options, and standards-compliant code to boot.

Themes

If your existing site uses FrontPage themes, it will still have the theme applied when you open it in Expression Web. However, the Page Banner and Link Bar Web components do not exist in Expression Web, and themes relied heavily on those components for the design, structure, and navigation of a Web page. In addition, FrontPage Navigation view, an essential feature for adding new pages to the Link Bar and applying the Page Banner to those pages, doesn't exist in Expression Web. So, while you can continue to edit your existing pages that have themes applied, you can no longer add new pages to the Link Bar Web component, nor add a Page Banner Web component to new pages. Given the limitations of FrontPage themes both in FrontPage and in Expression Web, you may want to consider creating your own theme by designing your own styles from scratch.

The good news is that if you have been using FrontPage themes, you have actually been using style sheets without realizing it. Themes are applied by attaching a style sheet to the page. You can view and even edit the style sheet from within both FrontPage and Expression Web. These style sheets are very complicated, however, and even a seasoned CSS developer would have a hard time wading through it all. Consider upgrading your Web site from themes to CSS by applying a new style sheet to your site and learning how to edit and manage styles in Expression Web. For an introduction to CSS, see Appendix A: Using Expression Web to Create Styles.

If you want to move away from FrontPage themes, first remove the current theme. Open your site in FrontPage. Press and hold the CTRL key and then, in the **Folder List**, click all of the pages to which a theme has been applied. Click **Format**, and then click **Themes**. In the **Themes** task page, click the arrow to the right of **No theme**, and then click **Apply to selected pages** to remove the theme(s) assigned to those pages.

You can now try the CSS style sheet templates included with Expression Web. With your Web site open, on the **File** menu, point to **New**, and then click **Page**. On the **Page** tab, click **Style Sheets**, and then click any of the style sheets other than the style sheet entitled **Blank**. After

you save the new style sheet with a new name, you can then apply it by dragging the style sheet onto the HTML page while it is open in Design view.

You can also create your own style sheet that functions in a manner similar to a theme. In the following example, you will create a simple style sheet to help get you started. The CSS code below defines the page background color, default font, and default text colors, and then provides specific formatting for headings and links. For more information on how to create alternatives to the Page Banner and Link Bar Web components, see the section entitled Link Bars and the section entitled Page Banner Included Content.

In the following example, you are going to create a new style sheet and new default Web page, and then apply the style sheet to the Web page.

To create the style sheet:

1. In the practice Web site you created for the page banner, in the **Folder List**, double-click on the style sheet that you created to either open it or bring it to the front. If the **Folder List** isn't visible, on the **View** menu, click **Folder List**, or press ALT+F1.
2. Copy the following code by selecting it and then pressing CTRL+C.

```
body {
/* this sets the background color of the page */
background-color: #ffffff; color: #414345;
/* this sets the default text color of the page */
/* this sets the default font and text size */
font: 80% Verdana, sans-serif;
}
h1, h2, h3, h4, h5, h6 {
color: #37526d; /* this sets the colors of the headings */
font-family: Georgia, 'Times New Roman', Times, serif;
/* this sets the font for all headings */
}
/* the styles below set font sizes and styles for specific headings */
h1 { font-size: 2em; }
h2 { font-style: italic; font-size: 2em; }
h3 { font-size: 1.7em; }
h4 { font-style: italic; font-size: 1.7em; }
h5 { font-size: 1.3em; }
h6 { font-style: italic; font-size: 1.3em; }
/* the styles below set link styles and colors */
a:link { color: #0a5baf; }
a:visited { color: #5883b0; text-decoration: line-through; }
a:active, a:hover { color: #c65d10; }
```

3. Place the insertion point below your existing page banner style, and then paste the code you just copied into the style sheet by pressing CTRL+V.
4. On the **File** menu, click **Save**.

To create the sample HTML:

1. In the **Folder List**, double-click Default.htm either to open it or to bring it to the front.
2. Copy the following text by selecting it and then pressing CTRL+C.

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
```

```
<h6>Heading 6</h6>
<p>Normal text</p>
<p><a href="default.htm">Link color</a> |
<a href="about.htm">Hovered link color</a> |
<a href="photogallery.htm">Active link color</a> |
<a href="links.htm">Visited link color</a></p>
```

3. In **Code** view, place the insertion point after the H1 element from the page banner example, and then paste the text by pressing CTRL+V.
4. To apply the style sheet to the Web page, with Default.htm open in **Design** view, in the **Folder List**, click the style sheet you created and drag the style sheet onto the Web page. The resulting page should look something like the following illustration.

Note: The Hovered Link Color link, Active Link Color link, and Visited Link Color link will appear the same as the Link color link until you either hover over the link, click the link, or move the pointer away from the link.

Along with the page banner in the previous example, you have just created your first theme!

Shared Borders

If your site already uses Shared Borders, you can continue to use and edit them in Expression Web. However, shared borders were already being phased out in FrontPage 2003 in favor of Dynamic Web Templates. With Expression Web, you have the option of transitioning to a newer, more enhanced version of Dynamic Web Templates. If your site is hosted on a server that supports ASP.NET, you can also use ASP.NET Master Pages.

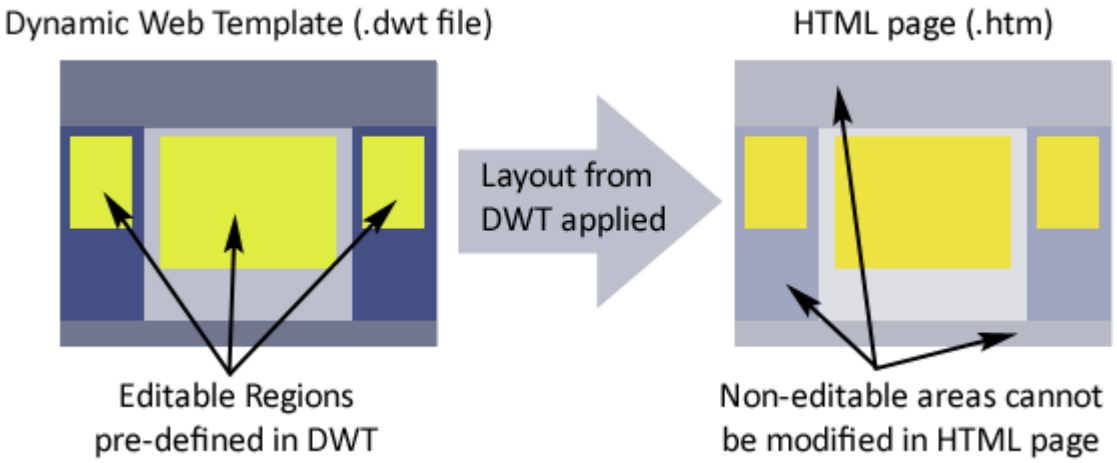
Dynamic Web Templates

Dynamic Web Templates (DWT) are template pages with a .dwt extension. You can create a single page that contains the layout and design for your entire site, and then attach that template to every other page in your site. In addition, the content in the DWT can only be edited in the DWT file itself, meaning that you can easily create a consistent look and feel throughout your entire site, and easily edit the content of each individual page, while also protecting the overall design from accidental edits on the page.

The DWT can be created and edited like a normal Web page in Expression Web, so you can create a master layout—the header, footer, content columns, navigation, copyright statement, and any other content that that you want to include in each of your Web pages. You can also create multiple DWTs for a single site. For example, you might want a two-column template, a three-column template, and perhaps a special template for a product detail page.

Instead of adding content as you would on a normal Web page, you define editable regions within each DWT. For instance, you might add an editable region to the content column for your main content, a column to the left, and another column to the right column for additional content such as submenu navigation or ads. You then create content in the page that is attached to the DWT. The editable region in the attached page is the only region that can be modified. This keeps the template design from being accidentally modified from page to page, and it also allows DWTs to be interchangeable. You can attach a different DWT to a page for a different design—for example, you can take an HTML page with a two-column DWT attached, and attach a three-column DWT instead. The page layout will automatically update without changing the existing content in the editable region, and all without any painful copying, pasting, and recoding.

Another benefit of using DWTs is that global changes are much easier to make. If you need to make a change, you can make the change in one place, the DWT file. After saving your changes, Expression Web prompts you to apply the change to all the related HTML pages that use that DWT.



You can create a DWT by creating a new empty page (on the **File** menu, click **New**; in the **New** dialog box, click **General**, and then click **Dynamic Web Template**), but you may find it easier to start with an existing HTML page that already has your basic layout in place. You can then save your page as a DWT by clicking **File**, then clicking **Save As**. In the **Save As** dialog box, in the **File name** box, type a name for your DWT. In the **Save As Type** dialog box, click **Dynamic Web Template**.

If you have been following the examples throughout this paper, you can easily create a basic DWT in the practice Web site you created earlier.

To create the DWT:

1. On the **File** menu, click **New**. In the **New** dialog box, on the **Page** tab, click **General**, and then click **Dynamic Web Template**.
2. Copy the following code by selecting it and then pressing CTRL+C.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html dir="ltr" xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- #BeginEditable "doctitle" -->
<title>My first DWT</title>
<!-- #EndEditable -->
<link rel="stylesheet" type="text/css" href="styles.css" />
</head>
<body>
<div id="top">
<div id="container">
<p><a href="default.htm">Home</a> |
<a href="default.htm">About Me</a> |
<a href="default.htm">Photo Gallery</a> |
<a href="default.htm">Contact Me</a></p>
<h1 id="pageBanner">Page Banner</h1>
<div id="left">
<p>Left column content here.</p>
</div>
<div id="right">
<p>Right column content here.</p>
</div>
</div>
</div>
</body>
</html>
```

3. In **Code** View, press CTRL+A to select all of the code in the page, and then press CTRL+V to replace the existing code with the code you just selected.
4. On the **File** menu, click **Save**. In the **Save As** dialog box, in the **File** name box, type a name for your template and then click **OK**. Note that the new file in the **Folder List** has a .dwt extension.

To create the editable regions:

1. Locate and then select the placeholder text. In this case, the placeholder text is **Left column content here**. Be sure to select the entire element. For example, select the **Left column content here** text, but also select the start <p> tag and the end <p> tag as in the following example:

```
<p>Left column content here</p>
```

2. Right-click the selected text and then click **Manage Editable Regions**. In the **Editable Regions** dialog box, in the **Region name** box, type a name for the region (for example, **left**), and then click **Add**.
3. Repeat steps 1 and 2 for the placeholder text **Right column text**.

4. If you have Visual Aids selected (on the **View** menu, click **Visual Aids**, and then click **Show**), the editable regions are surrounded by an orange border with a tab at the top displaying the name of the editable region.

When you look at the HTML in Code view, you'll see the editable regions defined within comment tags:

```
<!-- #BeginEditable "left" -->
<p>Left column content here</p>
<!-- #EndEditable -->
```

The area between the BeginEditable and EndEditable comment tags is the area where content unique to each page, content not repeated by using the DWT, is located.

You can add as many editable regions as you would like. For example, you may want to add an editable region within each column for images, a region for an article title, or a region for a submenu.

In this section, you'll create the style sheet for your DWT:

1. In the **Folder List**, double-click the style sheet to open it or bring it to the front. Copy the following code by selecting it and pressing CTRL+C:

```
/* the styles below create the columns */
#container { width: 700px; background: #ffffff; padding:10px; }
#left { width: 200px; float: left; background: #ffffff; }
#right { width: 500px; float: right; background: #ffffff; }
```

2. Place the insertion point below your existing styles, and then paste the code you just copied into the style sheet by pressing CTRL+V.
3. Open your DWT in **Design** view. In the **Folder List**, click the CSS file you just modified and drag it onto the DWT page.

Now that you have created a DWT, you can create a new page based on that DWT.

1. On the **File** menu, point to **New**, and then click **Create from Dynamic Web Template**.
2. In the **Attach Web Template** dialog box, click the DWT that you just created, and then click **Open**. Expression Web alerts you that a file has been updated.

You are now ready to add content. Note that when you move your pointer over the page, when the pointer isn't over an editable region, it appears as a circle with a line through it. This lets you know that this section of the page is not editable. When you hover over the editable region, the insertion point becomes active, and you can add and delete text in the region.

You can also attach a Dynamic Web Template to a page that already has content in it. In the following example, you are going to paste some content into the page, attach a Dynamic Web Template, and then map the content to an existing editable region.

1. Click **File**, point to **New**, and then click **Page** to create a new page.
2. Copy the following content by selecting it and then pressing CTRL+C.

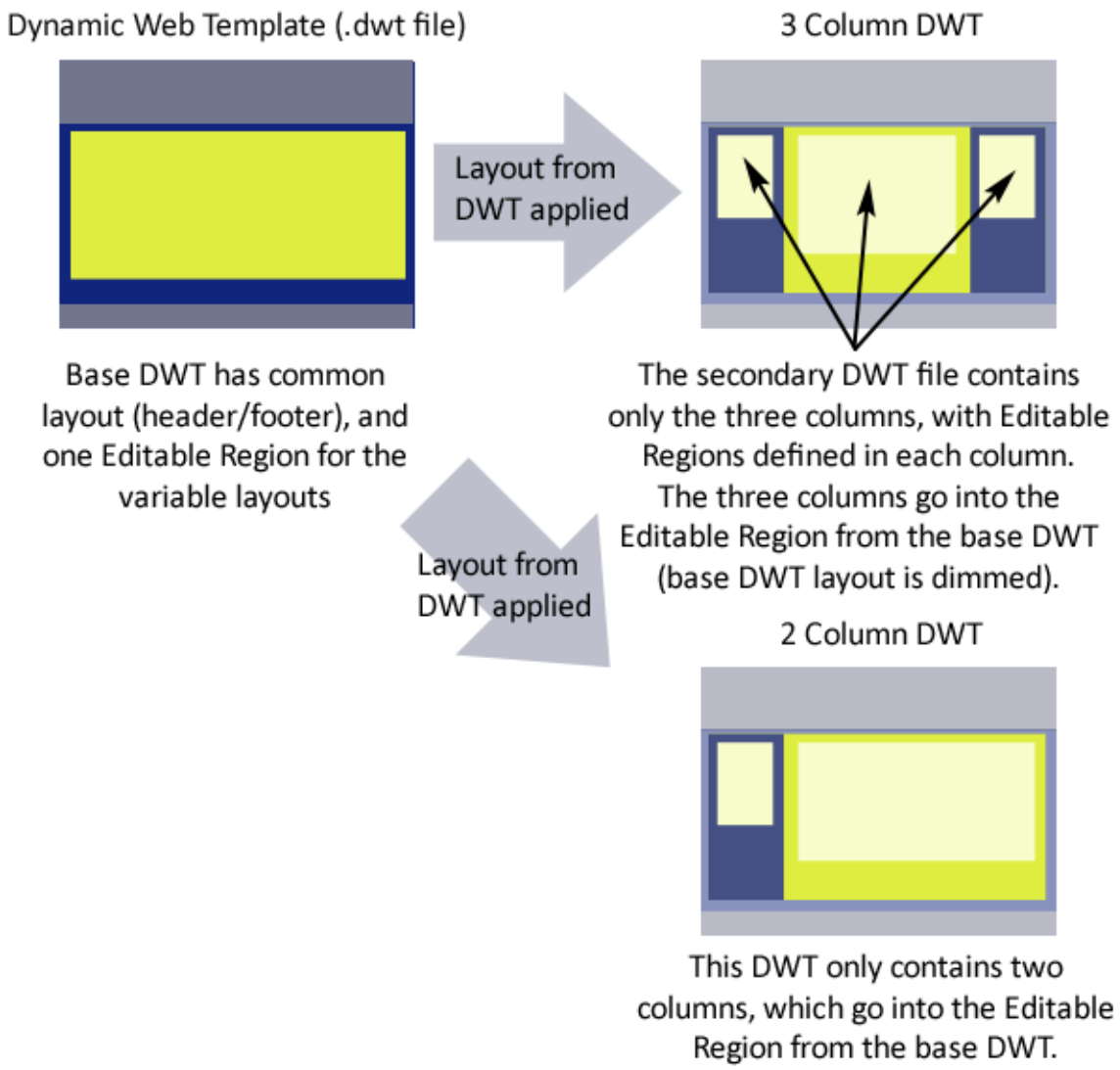
```
<p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Sed vel
nisi. Aenean et nulla fringilla tellus ullamcorper aliquam. Proin
venenatis. Aliquam erat volutpat. Morbi id magna nec erat posuere cursus.
Nullam facilisis, lorem eu lobortis pellentesque, ante turpis tincidunt
leo, faucibus cursus est nibh ac mi.</p>
```

```
<p>In at nunc. Morbi consequat ornare nibh. Curabitur nisl. Sed sapien.
Nulla porttitor ullamcorper magna. Donec urna. Mauris condimentum lorem
non massa. Aliquam euismod leo. Nam lobortis vehicula turpis. Sed
adipiscing, ante at pharetra hendrerit, nisi ipsum ultricies risus, id
euismod odio urna vel nulla. Nunc erat neque, tempor nec, congue ac,
malesuada vulputate, enim. Ut porta congue eros.</p>
```

3. In **Code** view, place the insertion point between the start and end body tags, and then press CTRL+V to paste the content into the page.
4. On the Format menu, click Dynamic Web Template, and then click Attach Dynamic Web Template.
5. In the **Attach Web Template** dialog box, click the DWT that you created, and then click **Open**. Expression Web alerts you to move the text into an editable region. Click **Yes**.
6. In the **Match Editable Regions** dialog box, under **Other regions on this page**, there are two columns. The first column is the region that contains the content displayed in the current page, the BODY element. The second column is the first editable region in the DWT.

For this example, we are going to match the BODY content with the Right editable region. Click (Body), and then click Modify. In the New Region box, click Right, and then click OK. Click OK again, and Expression Web alerts you that a page has been updated. The text now appears in the right column editable region.

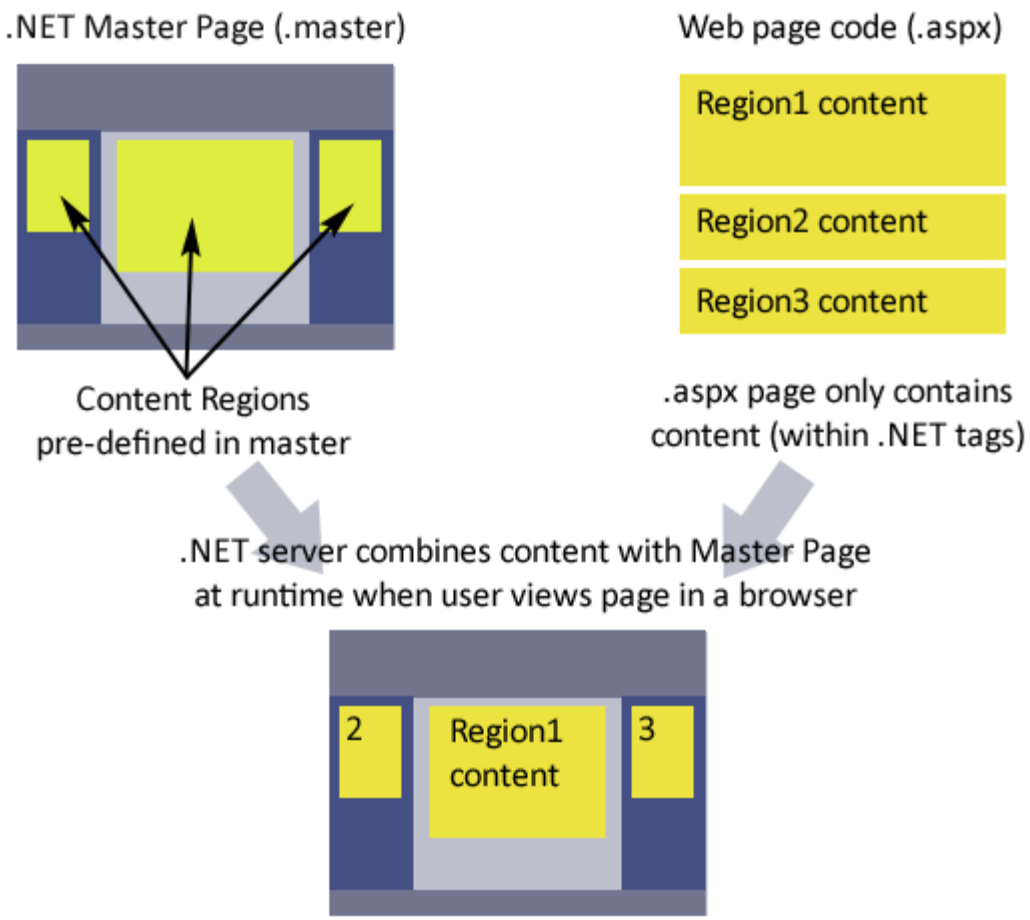
Expression Web has taken DWTs to a new level by adding the ability to have nested Dynamic Web Templates. If you used DWTs in FrontPage 2003, you may have needed several different DWTs for a Web site—for example, a two-column DWT, a three-column DWT, and perhaps others. You may have also used Include Pages to further share content between the DWTs; for example, each DWT may have used an Include Page for a common header, footer, and navigation area. In Expression Web you can define a base DWT—for example, with just the header, footer, and navigation—and then create additional DWTs containing only the column layouts. You can then attach the base DWT to the column DWTs. By attaching the main DWT to the different column layouts, you still only need to make design changes in one location, while creating multiple DWTs by nesting the column layout DWTs within the main DWT.



ASP.NET master pages are similar to Dynamic Web Templates in that you create a template, called a master page, with defined content regions similar to editable regions in DWTs. The main differences between the two are:

- ASP.NET Master Pages require you to have your site on a server with ASP.NET 2.0 installed in order to render the ASPX pages, while Dynamic Web Templates create HTML pages that can be displayed on any server.
- Dynamic Web Templates render as HTML pages. For example, if you were to open the HTML page in a text editor, you would be able to see all of the code. ASP.NET pages render “on the fly” from the server. Opening an ASPX page in a text editor would show you some code that describes which master page is attached, but you won’t actually see the code from the master page in the ASPX page.

When you edit a DWT file, Expression Web prompts you to apply the changes to all the HTML pages that use the DWT. This can take a bit of time if you have attached the DWT to several HTML pages. You then have to republish all of the changed HTML pages. Because ASPX pages are rendered on the fly, updating the master page does not require updating the pages to which the master page has been applied.



Creating an ASP.NET master page is similar to creating a Dynamic Web Template. Again, you may find it easier to start with a page that has your base layout already created. You can use the same layout as you used for the Dynamic Web Template.

1. In your practice site, on the **File** menu, click **New**. In the **New** dialog box, on the **Page** tab, click **General**, and then click **Master Page**. Click **OK**.

2. Copy the following code by selecting it and then pressing CTRL+C.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%@ Master Language="C#" %>
<html dir="ltr" xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>My First ASP.NET Master Page</title>
<asp:ContentPlaceHolder id="head" runat="server">
</asp:ContentPlaceHolder>
</head>
<body>
<form id="form1" runat="server">
<div id="top">
<div id="container">
<p><a href="default.htm">Home</a> |
<a href="about.htm">About Me</a> |
<a href="photos.htm">Photo Gallery</a> |
```

```
<a href="links.htm">Contact Me</a> |
<a href="mailto:someone@somewhere.com">Contact Me</a></p>
<h1 id="pageBanner">Page Banner</h1>
<div id="left">
<p>Left column content here.</p>
</div>
<div id="right">
<p>Right column content here.</p>
</div>
</div>
</div>
</form>
</body>
</html>
```

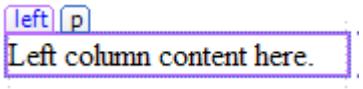
3. In **Code** view, press CTRL+A to select all of the code in the page, and then press CTRL + V to replace the existing code with the code you just selected.
4. On the **File** menu, click **Save**. In the **Save As** dialog box, in the **File** name box, type a name for your template and then click **OK**. Note that the new file in the **Folder List** has a .master extension.

To create the content regions:

1. Locate and then select the placeholder text. As with the DWT example, the placeholder text is **Left column content here**. Be sure to select the entire tag. For example, select the **Left column content here** text, but also select the start <p> tag and the end <p> tag as in the following example:

```
<p>Left column content here</p>
```

2. Right-click the selected text and then click **Manage Microsoft ASP.NET Content Regions**. In the **Manage Content Regions** dialog box, in the **Region name** box, type a name for the region (for example, **left**), and then click **Add**.
3. Repeat steps 1 and 2 for the placeholder text **Right column text**.
4.
5. If you have Visual Aids showing (on the **View** menu, click **Visual Aids**, and then click **Show**), the editable regions are surrounded by a purple border with a tab at the top displaying the name of the content region.



When you save the page, Expression Web adds a contentplaceholder control called HEAD in the head of your document. A contentplaceholder control acts much like an editable region in a DWT. However, in a master page, an editable region is called a content region.

When you look at the HTML in Code view, you'll see the content regions defined between ASP.NET content placeholder tags:

```
<asp:ContentPlaceHolder runat="Server" id="left">
<p>Left column content here.</p>
</asp:ContentPlaceHolder>
```

The area between the **ContentPlaceHolder** control start and end tags is the area where content unique to each page, content not repeated by using the ASP.NET master page, is located.

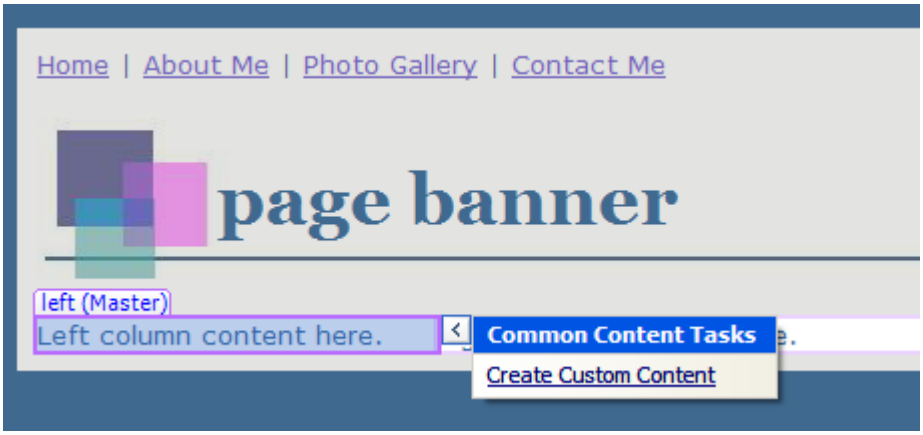
You can add as many content regions as you would like. For example, you may want to add a content region within each column for images, a region for an article title, or a region for a submenu.

Now that you have created your master page, it is time to apply the style sheet. Switch to Design view. Click the CSS page you modified earlier, and then drag it onto the page to apply the styles.

Now you can create a new page based on that master page.

1. On the **File** menu, point to **New**, and then click **Create from Master Page.**
2. In the **Select a Master Page** dialog box, click **Specific Master Page,** and then click **Browse**. In the **Select a Master Page** dialog box, click the master page that you just created, and then click **Open**. Click **OK**. A new page that has been created from your master page opens.

You are now ready to add content. Note that when you move your pointer over the page in Design view, when the pointer isn't over a content region, it appears as a circle with a line through it. When you try to add content to the page, you'll notice that the content regions are initially locked. To add custom content to the content regions, you'll first have to unlock the content regions. Click the arrow that appears in the upper right corner when you move the pointer over a content region. In the **Common Content Tasks** list, click **Create Custom Content**. You can now add content to the content region.



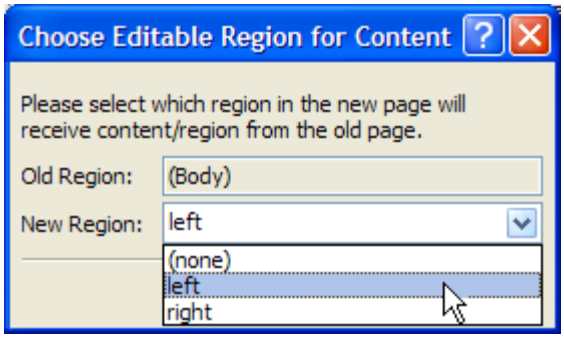
You can also attach master pages to existing HTML pages. The process is similar to attaching a Dynamic Web Template.

1. Click **File**, point to **New**, and then click **ASPX** to create a new ASPX page.
2. Copy the following content by selecting it and then pressing CTRL+C.

```
<p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Sed vel
nisi. Aenean et nulla fringilla tellus ullamcorper aliquam. Proin
venenatis. Aliquam erat volutpat. Morbi id magna nec erat posuere cursus.
Nullam facilisis, lorem eu lobortis pellentesque, ante turpis tincidunt
leo, faucibus cursus est nibh ac mi.</p>
<p>In at nunc. Morbi consequat ornare nibh. Curabitur nisl. Sed sapien.
Nulla porttitor ullamcorper magna. Donec urna. Mauris condimentum lorem
non massa. Aliquam euismod leo. Nam lobortis vehicula turpis. Sed
adipiscing, ante at pharetra hendrerit, nisi ipsum ultricies risus, id
euismod odio urna vel nulla. Nunc erat neque, tempor nec, congue ac,
malesuada vulputate, enim. Ut porta congue eros.</p>
```

3. In **Code** view, place the insertion point between the start and end <body> tags, and then press CTRL+V to paste the content into the page.

4. On the **Format** menu, click **Master Page**, and then click **Attach Master Page**.
5. In the **Select a Master Page** dialog box, click **Specific Master Page**, and then click **Browse**. In the **Select a Master Page** dialog box, click the master page that you just created, and then click **Open**. Click **OK**.
6. In the **Match Content Regions** dialog box, there are two columns. The first column is the region in the current page. The second column is the first region in the master page. Click **(Body)**, and then click **Modify**. In the **New Region** list, click **Right**, and then click **OK**. Click **OK** again. The text now appears in the right column editable region.



You can apply either a DWT or master page to multiple pages in your Web site by clicking the files in your **Folder List** (press the CTRL key to select multiple pages). Once you have selected the page(s), on the **Format** menu, click either **Dynamic Web Template** or **Master Page**, and then click either **Attach Dynamic Web Template** or **Attach Master Page**. You will then be prompted to match editable or content regions for each page. Typically you will want to attach the template to a single page initially, to make sure that it looks the way that you want it to before attaching the template to multiple pages.

Other FrontPage Server Extension-based Features

There are other FrontPage features that relied on FrontPage Server Extensions that are not found in the **Insert Web Component** dialog box and which are not part of Expression Web. This section gives a quick summary of the discontinued components and a few alternatives.

Database Connections

FrontPage supported database creation and management by using the Database Interface Wizard. The Database Interface Wizard doesn't exist in Expression Web. Instead, data integration has been designed to use the latest Web technologies, allowing you to integrate data not only from a variety of databases but also from XML data files. As with other FPSE-based features, you can continue to manage your existing database connection in Expression Web. However, if you want to create new database connections, you may want to review the following resources:

- The Rich Data Presentation and Powerful Server Technology training videos available on the Microsoft Expression Web site.
- The Expression Web Help documentation on XML and Databases (with Expression Web open, press F1, and then click XML and Databases in the Table of Contents).
- The Database Tools tutorials on the OutFront Web site.

Comment

The Comment command (available on the **Insert** menu in FrontPage 2003) is a design-time component that made it easy to include comments directly in Design view. The comments did not appear in the browser. Many FrontPage users found this a helpful tool, either as a reminder to themselves, or as a way to leave instructions for others working on the same site. As a design-time component, the component does not rely on FrontPage Server Extensions.

If you have existing comments in your page, double-click the component to access the **Comment** dialog box. Since the component is not dependent on FrontPage Server Extensions, you can also use the same code to create new comments in Expression Web.

You can also use the code to create a code snippet for easy reuse later on.

To create the code snippet:

1. On the **Tools** menu, click **Page Editor Options**.
2. In the **Page Editor Options** dialog box, on the **Code Snippets** tab, click **Add**.
3. In the **Add Code Snippet** dialog box, in the **Keyword** text box, type a unique keyword for the code snippet (Comment for example), which you can later use to select the code snippet in the list of code snippets.
4. In the **Description** box, type a description to identify the code snippet. In the **Text** box, type or copy and paste the following code:

```
<!--webbot bot="PurpleText" PREVIEW="" -->
```

5. To insert the Comment code snippet:
6. In **Code** view, place the insertion point where you want the comment to appear, and then press **CTRL+ENTER.**
7. Select the **Comment** keyword from the list of code snippet keywords, and then press **ENTER**.

8. Double-click the **Comment** placeholder. In the **Comment** dialog box, type the comment. Click **OK**.

Conclusion

Expression Web is distinctly different from FrontPage, boasting powerful CSS management and editing tools, built-in standards compliancy, browser compatibility and accessibility checkers, run-time ASP.NET support, and both Dynamic Web Templates and ASP.NET master pages. The major differences between Expression Web and FrontPage all have to do with allowing Web developers to create compliant, scalable, and accessible Web sites that meet today's standards. Sophisticated CSS support and task panes make it easy for new users to learn and start using CSS, and for FrontPage users to move away from themes, link bars, and other FrontPage components that are no longer recommended in FrontPage and not available in Expression Web.

Expression Web was designed to allow FrontPage users to support, maintain, and improve their FrontPage Web sites. Most of the FrontPage Web components can be easily modified in Expression Web. This document has also provided alternatives for those FrontPage users seeking to move away from Web components and create sites standards-compliant Web sites. Expression Web's interface is similar to FrontPage, making it easy for FrontPage users to transition to Expression Web, but it comes with many new tools—and the ability to customize the interface layout—to help Web designers and developers work more efficiently.

FrontPage users who are interested in bringing their Web sites to higher levels of sophistication and standards compliancy will find Expression Web an invaluable tool for developing Web sites.

Appendix A

Using Expression Web to Create Styles

If you're encountering CSS for the first time, Expression Web makes it simple for you to start creating styles and style sheets. This section will cover some CSS basics and then show how to best use Expression Web to create, edit, manage, and apply styles.

CSS is a coding language that was designed to help define how HTML elements are displayed in a browser. By implementing CSS, you can separate your content (text, images, and other information) from the presentation (the design or "look and feel" of the page). By separating content from presentation, you are not only more closely following Web standards, but also enhancing compatibility, accessibility, and speeding the download times for each of your pages.

CSS can be implemented in three ways: inline, embedded, or externally.

Inline Styles

Inline styles are styles applied to the tag inline (directly to the selected text within the HTML document). Inline style definitions use the STYLE attribute. For example, you can set the background color of an entire paragraph to yellow from within the paragraph by using the following inline style:

```
<p style="background-color: #ffff00">Paragraph text</p>
```

Inline styles should be used only rarely. Not only do inline styles not separate the content from the presentation, but they also present maintenance concerns as future changes would require going into each instance of the inline style and modifying the properties in the code.

Internal or Embedded Style Sheets

Internal or embedded styles are styles that are located in the HEAD of the document and are available to be used through the entire page. For example, the following code specifies that all paragraphs on the page have a yellow background.

```
<style type="text/css">
p { background-color: #ffff00; }
</style>
```

External Style Sheets

External style sheets are separate files from content pages and can be linked to or imported into any Web page. This allows you to define a style for multiple pages in a Web site, or use multiple style sheets for a single page (for more information, see Customized Displays for Multiple Media Types) allowing for greater flexibility, easier maintenance, and true separation of content from presentation.

Creating a CSS style definition

A CSS style definition has three parts: a selector (what you're applying the style to, for example, a paragraph, or <p> tag), the property (what you want to define, for example, the font), and a value (the format, for example, a font family such as Verdana). The properties and values are contained within curly brackets.

```
selector {
     property1: value;
     property2: value;
}
```

The selector can be an element (P, H1, IMG, etc.), which defines properties for that tag everywhere it appears. The selector can also be a class, a style that is only applied to those elements to which you assign that specific class. A class name begins with a period, for example, **.redtext**, **.accent**, and **.price**. Classes can be used with several different elements on a page. For example, the **.redtext** class be applied to single or multiple words, entire paragraphs, headings, or specific sections on a page.

You can also use IDs, which in CSS files are identified with a pound sign (#). IDs should be unique, and should only be applied once on a Web page. For example, you might choose to create an ID selector called **#left_content** and apply it to a DIV (** **. Once you do that, you shouldn't use the **#left_content** ID anywhere else on the page. This practice of using an ID only once in a page is particularly useful as you get into the more advanced uses of CSS in combination with JavaScript.

After the selector comes a pair of curly brackets, which surround your list of properties. Each property is then set to a value. Properties are separated by semi-colons.

The cascading nature of CSS refers to two things: first, styles are applied ("cascaded") based on where they are located, meaning whether they are inline, embedded, or external. CSS also means that certain properties defined in a *parent* element will apply to *child* elements. For example, if you have the text color property defined in the **body** element, all the paragraphs, headings, and other text will pick up the text color property unless otherwise defined.

Apply Styles by Using the Formatting Toolbar

One of the easiest ways to begin creating styles is to simply use the formatting toolbar options. By default, Expression Web will create an internal style. For example, by selecting text, then changing the color and size of the text, Expression Web creates a new class (**.style1**) and inserts the style sheet code in the HEAD of the document at the top of the page:

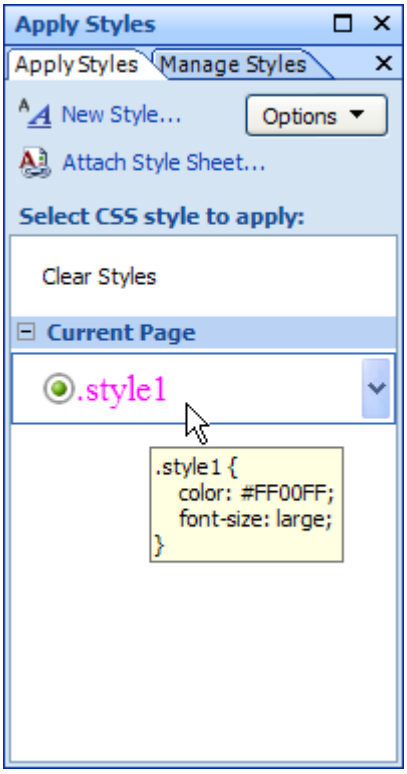
```
<style type="text/css">
.style1
{
  color: #FF00FF;
  font-size: large;
}
</style>
```

Compare the code above with the code that FrontPage generates by default:

```
<p>Example <font color="#FF00FF" size="4">text</font></p>
```

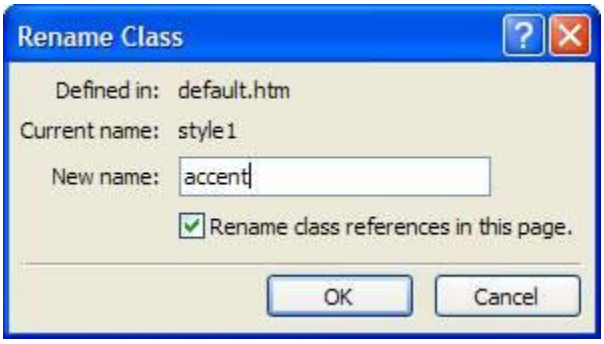
Applying a Style by Using the Apply Styles Task Pane

If you apply the same formatting options in another area of the page, Expression Web is smart enough to recognize that it should use the same style. Instead of creating a style2 class, it will apply the style1 because it has all of the same values assigned to the same properties. However, you can save yourself time by simply applying that style from the **Apply Styles** task pane. (If the **Apply Styles** task pane isn't visible, on the **Task Panes** menu, click **Apply Styles**.) The **Apply Styles** task pane shows all of the available styles in a document. After creating the new style, style1 now appears in the **Apply Styles** task pane. To apply the same formatting in a different area of the page, you can just select the other text and then click style1 in the **Apply Styles** task pane to apply it.



Renaming styles

You can also rename styles to make them more descriptive. This can be done in either the **Apply Styles** task pane or the **Manage Styles** task pane. Right-click the style name and choose **Rename class** to bring up the **Rename Class** dialog box. In the **Rename class** dialog box, you may type the new, more descriptive style name (no spaces or special characters, however). Selecting the **Rename class references in this page** checkbox will automatically fix the current code to use the new style name.



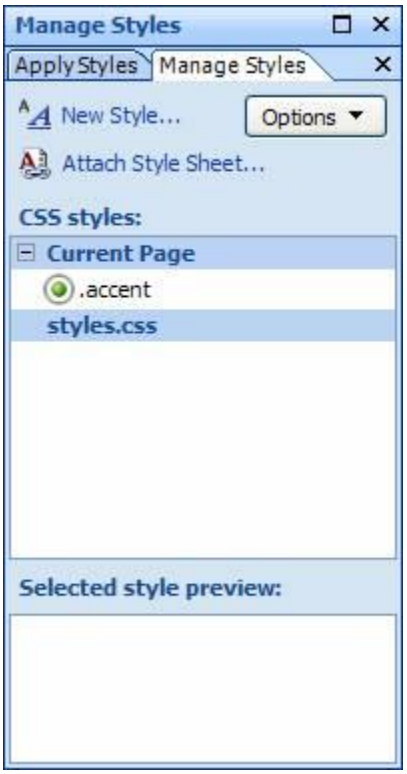
Moving a Style to an External Style Sheet

Internal styles only apply to a single page. If you want to use a style on multiple pages, you should move the style to an external style sheet. Expression Web makes moving style definitions and managing styles very easy from within the Manage Styles task pane.

If you do not already have an external style sheet for your site, you will have to create one and attach it before moving the style. On the **File** menu, point to **New**, and then click **CSS** to create a new CSS page. Save the page by clicking **File**, then clicking **Save**. In the **Save As** dialog box, in the **File name** box, type a name for your CSS file and then click **OK**.

Return to the Web page that contains the internal styles. In the **Manage Styles** task pane, click **Attach Style Sheet**. In the **Attach Style Sheet** dialog box, click **Browse**. In the **Select Style Sheet** dialog box, click the style sheet that you just created and then click **Open.**

If you want to attach this new style sheet to all of the HTML pages in your site, select **All HTML pages**. Otherwise keep **Current page** selected. The **Attach as** option allows you to attach the style sheet by using the <link> tag or by using an @import statement. Since you are just starting, select the **Link** option. Click **OK**. You will see the <link> tag added to the HTML code in your page. In both the **Manage Styles** and **Apply Styles** task panes, you'll see your style sheet under **CSS styles**, indicating that it's available and attached to the current page.



At this point, it's simple to move a new style from your current page into the external style sheet. Simply click on the style, drag it from the **Current Page** section to the external style sheet section (Styles.css). This removes the style code from the current page and adds it to the CSS page. Now you can use the class in any Web page that has the style sheet attached.

Organizing Styles

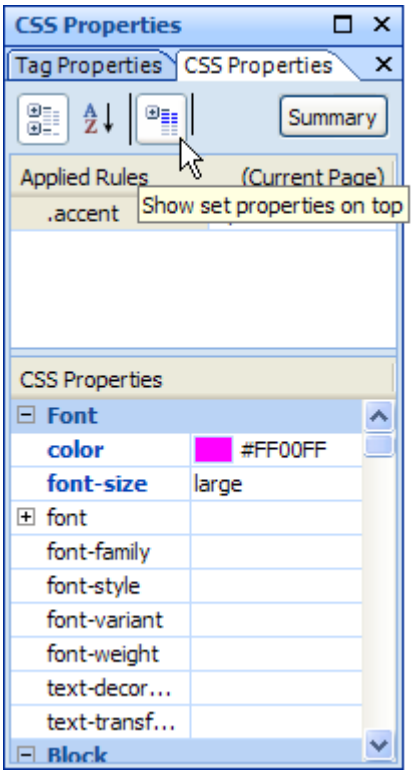
The Manage Styles task pane also allows you to organize your style sheets by making it easy to reorder styles. When you have multiple definitions in a style sheet, the Manage Styles task pane displays the styles in the same order that they appear in the actual CSS code. You can drag and drop the styles within the Manage Styles task pane to reorder the styles, and the corresponding style sheets will be updated.

Editing Styles

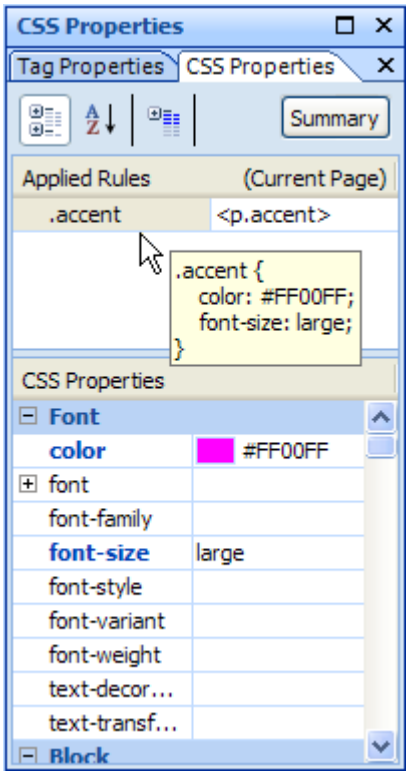
There are two ways that you can edit styles, and both work essentially the same way. The first is to right-click the style in the **Manage Styles** or **Apply Styles** task pane and then click **Modify Style** to open the **Modify Style** dialog box. This dialog box shows you all the different properties available and allows you to assign a value to each one. While most of the property

names are descriptive and intuitive, prior knowledge of CSS is helpful for knowing how all of the different properties can be used. If something isn't immediately obvious, you can experiment with the setting to see what it does, or look up that specific property and learn about it from an online resource such as the W3C's CSS school.

You can also edit the style by using the CSS Properties task pane. When you select an area on your page that has a style applied, the CSS Properties task pane shows all of the properties that have been set in bold blue text with the value showing in the right column. In the illustration below, the third button has been selected, allowing set properties to show at the top of the list of properties. You can edit the style by adding values for other properties.



The CSS Properties task pane shows all of the styles and properties that have been set, including the application of any cascading styles from parent elements. You'll want to make sure that you have selected the rule that you want to edit when you edit styles in the CSS Properties task pane. By selecting **.accent** in the **Applied Rules** list, you can modify that specific style definition.



Creating New Styles from Scratch

The formatting toolbar is an easy way to create simple styles works when working with text. However, if you want to assign several values to a style, it may be easier to click **New Style** in the **Apply Styles** or **Manage Styles** task panes. This brings up the **New Style** dialog box. In the **Selector** box, click on **Inline Style** to create an inline style, an HTML tag, or type a new class or ID selector name directly into the box. Then, in the **Define in** box, click **Current page**, **New style sheet**, or **Existing style sheet**. The rest of the dialog box is identical to the **Modify Style** dialog box, where you to assign values to the CSS properties to create a new style.

Learning about styles

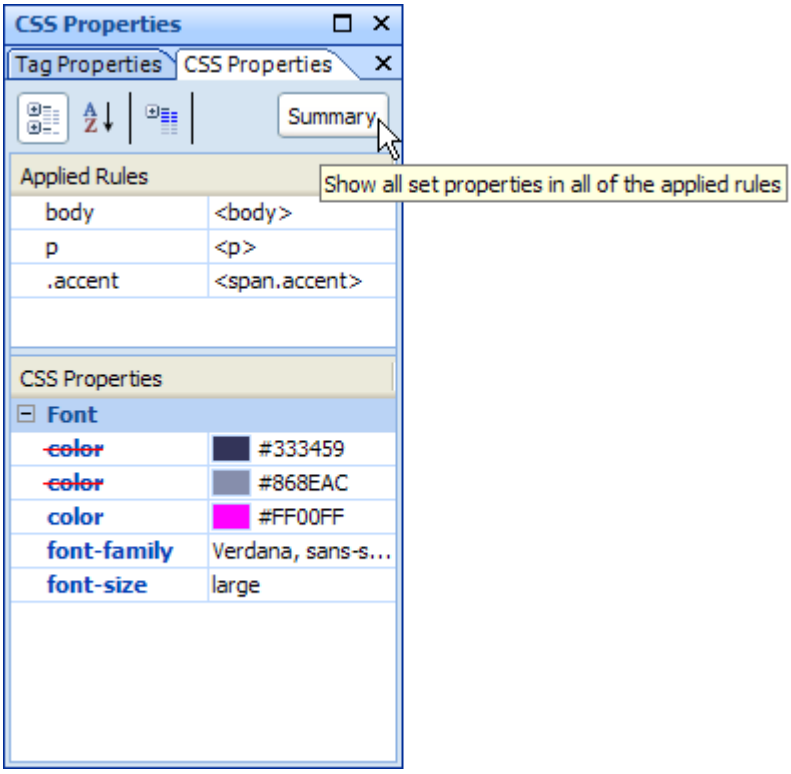
The sophisticated CSS tools and functionality of Expression Web make it easy to learn CSS as you go.

Move the cursor over the style name in the **Manage Styles** or **Apply Styles** task panes to bring up a screen tip displaying the CSS ScreenTip for that selector. The same ScreenTip appears when you move the cursor over a rule in the **CSS Properties** task pane. As you view the CSS code, you'll begin to get familiar with the various style properties, as well as which values are allowed and their purpose.

The CSS Properties task pane can also help you understand how different style rules relate to each other and which properties cascade or are redefined by rules. When you click the **Summary** button, the defined properties from the relevant rules are listed and unset properties are hidden from view.

The following illustration shows the variety of properties that apply to a specific area on a page as displayed in the CSS Properties task pane. By hovering over the rule in the **Applied Rules** list, you can view the CSS and see that the BODY element has a dark gray color (#333459) set for text. The P element has a light gray color (#868EAC) set for text, which

overrides the <body> color rule. To demonstrate this, in the **CSS Properties** task pane, Expression Web shows the dark gray color rule with a line through the property name, allowing you to see that the property has been overridden. The .accent class has a bright pink color (#FF00FF) set, which overrides the <p> color rule. Both of the other colors have been overridden by .accent, so both have lines through their property names. The **font-family** property set in the <body> rule has not been overridden, however, so it cascades and applies to the current selected text. Clicking **Summary** and viewing the CSS code screen tips allows you to view how the different rules relate to each other.



CSS is a powerful tool that allows you to format navigation links in a variety of ways. With CSS, you can:

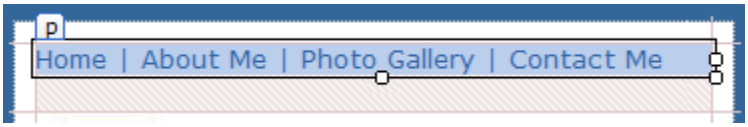
- Play with border colors and background colors for a 3D push button effect.
- Create a tab effect using a combination of borders and background colors
- Display list items inline or float links to the left to create horizontal navigation buttons.
- Use a background image in the link style for a graphic button effect. (This allows you to further simulate a FrontPage theme navigation button)

Appendix B

Using CSS to Style Links

This section explains how to apply basic font formatting to text links in a P, DIV, or TD element by using CSS. To apply formatting to links in a bulleted list, see Appendix C: Formatting a Bulleted List of Links with CSS.

To follow this example, your text links should be inside of a DIV element ( ), P element (<p>), or other block-level element such as a table cell (<td>). Once you have created your links, click the corresponding element tab (in the following example, the <p> tag) or the element name in the breadcrumb trail to select the entire element.



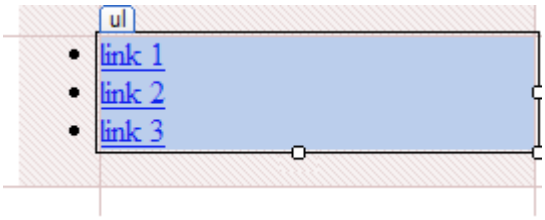
1. In the **Apply Styles** or **Manage Styles** task pane, click **New Style**.
2. In the **New Style** dialog box, in the **Selector** box, click **a:link**. This will allow you to set properties for the text links when they are in their normal state.
3. Select the **Apply new style to document** box.
4. In the **Define in** box, click **New Style Sheet** (if you haven't already created one) or **Existing Style Sheet** (if you have already created one). If you selected **Existing Style Sheet**, in the **URL** box, click **Browse**. In the **Select Style Sheet** dialog box, click your style sheet, and then click **Open**.
5. Set the **font-family**, **font-size**, **font weight**, and other properties that you would like for the text in your link bar. This allows you to set properties for the text links when they are in their normal state. Click **OK**.
6. Repeat for steps 2 through 5 for **a:active**, **a:hover**, and **a:visited**. When you have defined styles for all four selectors, you can then view your links in the browser by saving your page and pressing F12.

Appendix C

Formatting a Bulleted List of Links with CSS

In this section, you are going to set values for several selectors and properties to achieve an interactive button effect.

Your text links should be inside of an UNORDERED LIST element (<ul>), with each link within <li> tags. Click inside the text links and then click **<ul>** in the breadcrumb trail to select the entire bulleted list.



1. In the **Apply Styles** or **Manage Styles** task pane, click **New Style**.
2. In the **Selector** box, type a period (.) and then a name for the new style (.listMenu, for example).
3. Select the **Apply new style to document** selection box.
4. In the **Define in** box, select an existing style sheet, or, if you don't have a style sheet attached yet, select **New style sheet**.
5. In the **Category list**, click **Box**. Make sure **Same for all** is selected for both **Padding** and **Margin**. In the **Padding** column, in the **Top** box, click the **Up** arrow to activate the **Unit** box, and then click the **Down** arrow to set the value to **0px**. Repeat for the **Margin** column.
6. In the **Category** list, click **List**. In the **List-style-type** list, click **none**. Click **OK** to create and then apply the style.

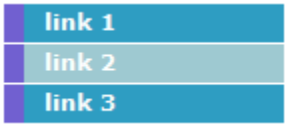
You'll notice that the bullets are gone and the list is flush with the left margin. Now you'll apply additional formatting to create the button look and feel. This set of steps defines the style for the links in their normal state.

1. In the **Apply Styles** or **Manage Styles** task pane, click **New Style**.
2. In the **Selector** box, type .listMenu a:link, .listMenu a:visited. Clear the **Apply new style to document selection** box.
3. In the **Category** list, click **Font**. In the **Font-family** list, click **Verdana**. In the **Font-size** list, click the **Up** arrow to activate the **Unit** box, and then set the font-size to **11 PX**. In the **Font-weight** box, click **Bold**. In the **Color** box, click **White** (**#FFFFFF**). Under **Text-decoration**, select **None**.
4. In the **Category** list, click **Block**. In the **Text-align** list, click **Left**.
5. In the **Category** list, click **Background**. In the **Background-color** box, type #2e9dc2.
6. In the **Category** list, click **Border**. Clear the **Same for all** checkboxes. In the **Border-style** list, in the **Left** box, click **Solid**. In the **Border-width** list, in the **Left** box, click the **Up** arrow to activate the **Unit** box, and then select **10 PX**. In the **Border-color** list, in the **Left** box, type #666798.

7. In the **Category** list, click **Box**. Clear the **Same for all** checkboxes. In the **Padding** list, in the **Left** box, click the **Up** arrow to activate the **Unit** box, and then select **10PX.** In the **Top** box, click the **Up** arrow to activate the **Unit** box, and then select **2 PX.** In the **Bottom** box, click the **Up** arrow to activate the **Unit** box, and then select **2 PX.**
8. In the **Category** list, click **Position**. In the **Width** box, click the **Up** arrow to activate the **Unit** box, and then select **120 PX**. In the **Height** box, click the **Up** arrow to activate the **Unit** box, and then select **15 PX**.
9. In the **Category** list, click **Layout**. In the **Display** list, click **Block**. Click **OK**.

You'll notice that the links are starting to look more like buttons. Now you'll apply additional formatting to define the style for the links in their active and hover states.

1. In the **Apply Styles** or **Manage Styles** task pane, click **New Style**.
2. In the **Selector box**, type .listMenu a:active, .listMenu a:hover. Clear the **Apply new style to document selection** box.
3. In the **Category** list, click **Background**. In the **Background-color** box, type #9ec9d1. Click **OK**.



Other resources

These sites may be helpful as you learn more about how to use CSS to format navigation links:

1. Taming Lists: An overview of formatting lists by using CSS.
2. Turning a list into a navigation bar: A blog entry with tutorials on how to use CSS to format lists.
3. Fast Rollovers Without Preload: A tutorial on how to use a background image to create theme-like navigation buttons.
4. Sliding Doors of CSS: A more advanced tutorial for using background images for scalable buttons.
5. Navigation Matrix Reloaded: An advanced CSS technique for using a background image for graphic buttons.