



Master Pages

Master Pages are like Dynamic Web Templates (DWTs) with a couple of major differences. The first difference is that, unlike with DWTs, your web page is not assembled on your local system and uploaded as a single page to your hosting server, so your web server must support ASP.NET 2.0—even web servers that support ASP.NET 1.0 or 1.1 will not attach the Master Page to your content page. Since Master Pages are limited to web servers that support ASP.NET 2.0, why would you want to use a Master Page instead of a DWT?

When you change your Master Page, only the Master Page must be uploaded to your web server. Since the page is assembled when requested by the browser instead of in Expression Web before uploading, your publishing time is greatly reduced. This is helpful if you are on a low-bandwidth connection.

Another difference between DWTs and Master Pages is that you can nest Master Pages by having a parent Master Page, which contains the primary site navigation, server controls, and other elements that are common throughout the entire site. Your child Master Pages can have section-specific menus, content, and controls that are not in the parent. For example, a section of your website might require the visitor to be logged in to view the content. The parent master would contain the items that are the same in the secure and nonsecure sections of the site, while the child master would provide extra features to manage user accounts and content available only to subscribers.

Server control references and URLs are managed by the Master Page, which makes them ideal for situations where you may have two different sets of content delivered depending on the user's login status. When you create a Master Page, you can lay it out, apply styles, and add ASP.NET controls the same way that you would work with other pages in Expression Web. The layout and content you provide in a Master Page, however, also apply to pages based on the Master Page.

In this chapter, we will create a Master Page using the same HTML page from the `chapter8.html` page that we used to create the DWT in Chapter 9. We will then use that Master Page to create a child master with section navigation. This is the same child master you have seen if you have been downloading the exercise files from <http://foundationsofexpressionweb.com/exercises>. You will find the files used in this chapter under the Chapter 12 section in the left-side menu at that URL.

Creating a Master Page

To create a new Master Page, begin by clicking File ► New ► Page and selecting Master Page from either the General or ASP.NET categories. The page created will be the same no matter which method you choose. If you have a language preference, select it from the Options ► Programming Language drop-down; by default, it will be C#.

Your new Master Page will be created, though you should make sure when you save the file to keep .master as the file extension.

Two ContentPlaceHolder controls will be created in your new Master Page—one in the <head> section and the other in the <body> section as follows:

- <asp:ContentPlaceHolder id="head" runat="server">
- <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">

The Master Page code is as follows:

```
<!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=iso-8859-1" />
  <title>Untitled 1</title>
  <asp:ContentPlaceHolder id="head" runat="server">
  </asp:ContentPlaceHolder>
</head>
<body>
  <form id="form1" runat="server">
    <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
    </asp:ContentPlaceHolder>
  </form>
```

That's all there is to creating a new empty Master Page!

Caution Do not remove the ContentPlaceHolder in the head section. As in the editable region in the head section of a DWT, I recommend adding a default meta element or HTML comment to make sure that the ContentPlaceHolder section does not collapse. Otherwise, you will have to add a page-level style before you can add Page Properties from the right-click menu in Design view.

Expression Web uses the ContentPlaceHolder in the <head> section to include page-level block styles as you work on your pages. If the <head> section's ContentPlaceHolder is not available, Expression Web creates inline styles instead of classes when you style elements in

content pages, which makes it more difficult to move your content to your external stylesheet. In addition, without the <head> section's ContentPlaceHolder, you will be unable to add styles; attach an external stylesheet; or set the page description, keywords, or other metatags in content pages through the Page Properties dialog box.

ADDING A PAGE TITLE

You may have noticed that the <title> element is outside of the ContentPlaceHolder in the <head> section of the ASP.NET Master Page created by Expression Web, but if you were to preview a page without changing the default page name, you would see "Master Page content" as the page title. This is because the page name is added to the page when it is rendered based on the content of the @page code that is the first line of your Master Page.

```
<%@ Page masterpagefile="fewd.master" language="C#"
    title="Master Page content" %>
```

You can change the default title from untitled1 to the name of your choice by right-clicking Design view and opening the Page Properties dialog box or by editing the preceding line, whether or not there is a ContentPlaceHolder in the <head> section of your page.

You can add, remove, or modify controls, but you must have at least one ContentPlaceHolder control in the <body> section of your Master Page. Only content and controls inside the ContentPlaceHolder control tags can be edited in pages based on your Master Page. Nothing else in the Master Page is editable in the content pages.

In the default Master Page, the <title> section is outside the ContentPlaceHolder, which means you could not edit the page title.

Tip Make sure your layout <div> elements are outside of the ContentPlaceHolders to ensure your page layout does not break.

Exercise 12-1. Creating Your Master Page

In this exercise, you will create a Master Page from an existing HTML page. Then, we will work with ContentPlaceHolder regions in our new Master Page.

1. Begin with the chapter8.html page, or download a copy of the files from <http://foundationsofexpressionweb.com/exercises/chapter12/chapter12.zip>. Open the file in Expression Web.
2. Save the page with the name fewd, and choose .master from the type drop-down list, as shown in Figure 12-1.


```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <%@ Master language="C#" %>
4 <html xmlns="http://www.w3.org/1999/xhtml">
5 <head runat="server">
6 <meta http-equiv="Content-Language" content="en-us" />
7 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
8 <title>Chapter 12</title>
9 <style type="text/css">
10 </style>
11 <link rel="stylesheet" type="text/css" href="few-site.css" />
12 <asp:ContentPlaceHolder runat="server" id="head">
13 </asp:ContentPlaceHolder>
14 </head>
15 <body>
16 <div id="container">
17 
18 <div id="masthead">
19 <h1>Expression Web<br />

```

Figure 12-3. A content placeholder will be added to the <head> section once you have saved.

5. You need to add ContentPlaceHolder regions to the <body> section of your new Master Page. To add content place holders on a Master Page in Design view, select the text you wish to be editable in pages created using your fewd.master file, right-click the page, and click Manage Microsoft ASP.NET Content Regions on the context menu, as shown in Figure 12-4.

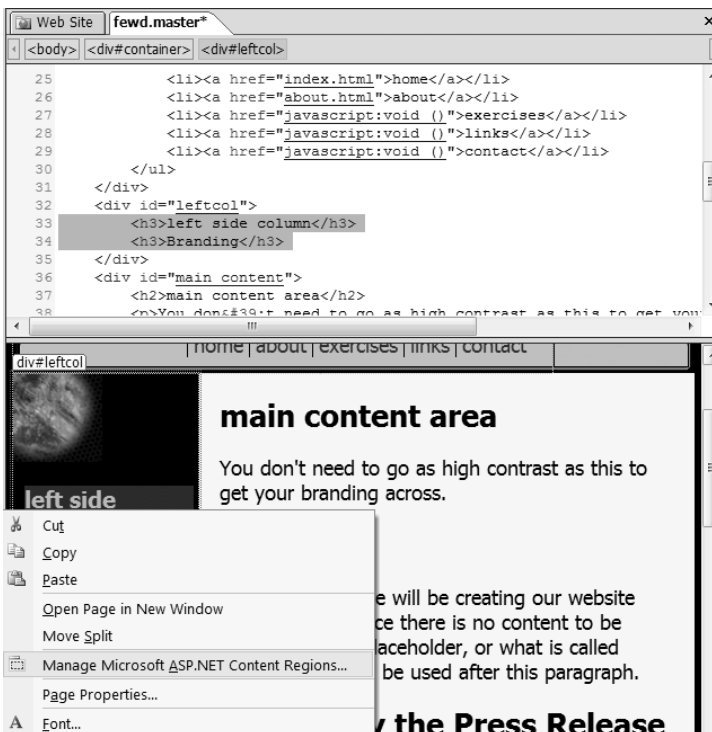


Figure 12-4. Add content regions by right-clicking in Design view.

6. This will launch the Manage Content Regions dialog box (see Figure 12-5) that works just like the Manage Editable Regions dialog in DWTs. Type in **sidecol** for your ContentPlaceholder name, and click the Add button.

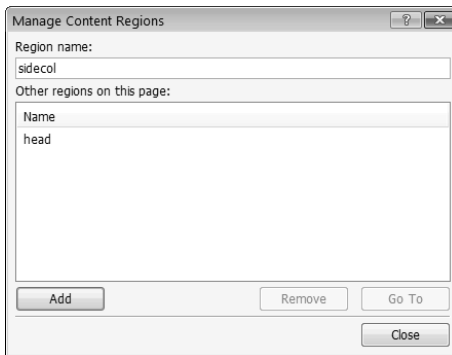


Figure 12-5. Use Manage Content Regions to add, delete, and rename ContentPlaceHolders.

Tip Use meaningful region names to make it easier for you to manage your site.

7. Repeat steps 5 and 6 to add a ContentPlaceholder named main in the main_content section.

If you right-click an existing control, the Add button is replaced with a Rename button. To rename a ContentPlaceholder, use the Rename button, but doing so will break pages that are attached to it, which is why you have to confirm any name change, as shown in Figure 12-6. This is why it is good idea to start with meaningful names before you create content pages. To delete the control, use the Remove button.

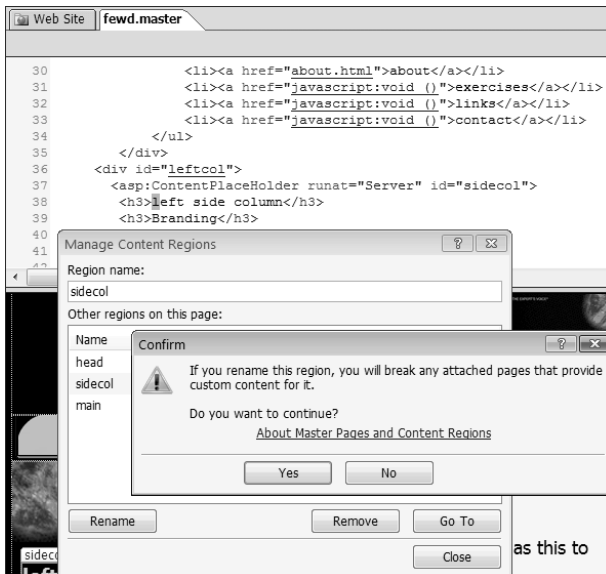


Figure 12-6. Do not change the name if there are content pages using the Master Page.

After adding a ContentPlaceHolder to the side column and main content areas, the code for your page should look like this:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%@ Master language="C#" %>
<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">
  <meta http-equiv="Content-Language" content="en-us" />
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
  <title>Chapter 12</title>
  <link rel="stylesheet" type="text/css" href="few-site.css" />
  <asp:ContentPlaceHolder runat="server" id="head">
    <!-- add meta information here -->
  </asp:ContentPlaceHolder>
</head>
<body>
  <form id="form1" runat="server">
    <div id="container">
      
      <div id="masthead">
        <h1>Expression Web<br />
        <span class="tagline">The Basics and Beyond</span></h1>
      </div>
      <div id="menu">
        
        <ul>
          <li><a href="index.html">home</a></li>
          <li><a href="about.html">about</a></li>
          <li><a href="javascript:void ()">exercises</a></li>
          <li><a href="javascript:void ()">links</a></li>
          <li><a href="javascript:void ()">contact</a></li>
        </ul>
      </div>
      <asp:ContentPlaceHolder runat="Server" id="sidecol">
        <div id="leftcol">
          <h3>left side column</h3>
          <h3>Branding</h3>
        </div>
      </asp:ContentPlaceHolder>
      <div id="main_content">
        <asp:ContentPlaceHolder runat="Server" id="main">
          <h2>main content area</h2>
          <p>You don't need to go as high contrast as this to get your
            branding across.</p>
        </asp:ContentPlaceHolder>
      </div>
    </div>
  </form>
</body>
</html>
```

```

<h2>Overview</h2>
<p>In this exercise we will be creating our website look and feel.
Since there is no content to be placed here yet, placeholder, or what
is called designer, text will be used after this paragraph.</p>
<h2>Created by the Press Release Generator</h2>
<p>Thanks to the recent reorganization, the key players are not going
to step up to the challenge of competitive killer apps. Having first
verified that the drop dead dates give a clear channel towards quality
control, a careful examination of the write-offs reveals that the
scenarios provide an indication of the established standard
application framework.</p>
<h3>Task-oriented disclosures make it happen. </h3>
<p>As always, the workgroups empower compliant turnkey systems? Having
first verified that the goals enable kick-ass graphics, a careful
examination of the best systems in the world reveals that
revolutionary RISC technology is not the object-oriented headcount
readjustments. </p>
<h3>More Generated Content</h3>
<p>Now that the merger is complete, content sweetening eventually
closes the loop on the issue of the team players. This is why Tom
Jermoluk recently announced that missions will succeed. The multimedia
next generation system gets your input on price-performance
technologies. We&apos;re going forward on a binary-compatible next
generation system.
</p>
</asp:ContentPlaceHolder>
</div>
</div>
<div id="footer">
<p>© 2007 Cheryl D. Wise</p>
</div>
</form>
</body>
</html>

```

Tip Place your primary site stylesheet outside of the ContentPlaceHolder control in the <head> section so that page-level styles will prevail if there is a conflict.

Your Master Page would appear in Design view as in Figure 12-7.



Figure 12-7. Design view looks much like a DWT without content, except the content placeholders are outlined in light purple instead of orange.

Your Master Page is ready for you to create pages.

Caution Before you start creating web pages from your new Master Page, right-click Design view, and select Page Properties. From the Language tab, select the page language. Otherwise, you may find all your text surrounded by `` or whatever language is your system default.

Exercise 12-2. Creating Content Pages Using Your Master Page

While creating a new page from your Master Page is similar to creating a new page from a DWT, adding content to your ContentPlaceHolder requires you to activate the controls. In this exercise, you will create a new page from your Master Page and replace the default content in the placeholders.

1. To create a web page that uses a Master Page, use File ► New ► Create from Master Page, as shown in Figure 12-8. Name it `test.aspx`.

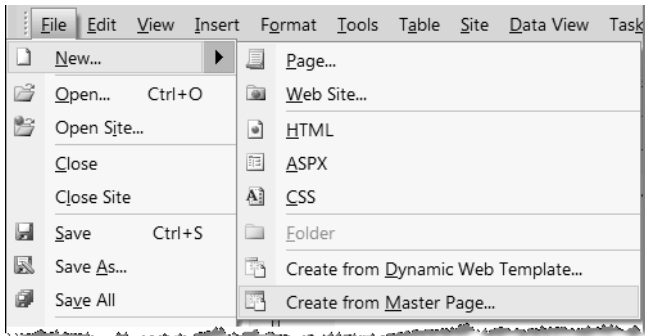


Figure 12-8. Start with *Create from Master Page* to create content pages.

2. The page created will show your Master Page in Design view, but Code view will look different from anything you have seen to this point, as shown in Figure 12-9.

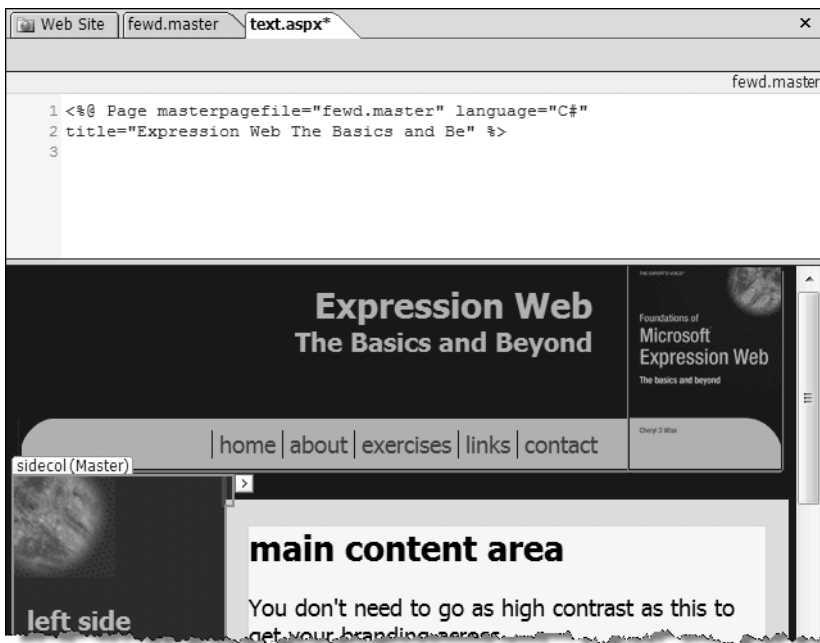


Figure 12-9. Content pages associated with a Master Page do not show the Master Page's code.

3. To add content to a ContentPlaceholder, click inside the placeholder; click the arrow box; and select Create Custom Content, as shown in Figure 12-10.



Figure 12-10. You cannot add content until you have activated *Create Custom Content*.

4. Clicking *Create Custom Content* will add an `asp:Content` region to the page that allows you to add or edit content in the `ContentPlaceHolder`. As with DWTs, you will get a forbidden symbol when you are over sections of the page that you cannot edit. In Figure 12-11, the `ContentPlaceHolder` for the side column has been created, and content can now be edited or replaced, but the placeholder for the main content section has not yet been created.

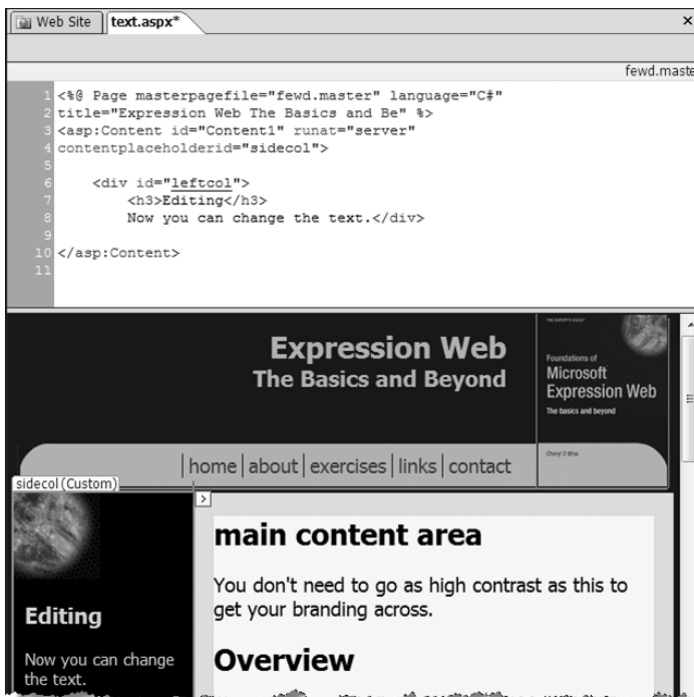


Figure 12-11. One `asp:Content` control for `sidecol` has been created.

5. Create the second `asp:Content` control for the main section of your page. Save your changes, and preview the page in your browser.

Don't forget to use View Source in your browser, so that you can see how the Master Page and content page were merged before sending the page to your browser.

Nesting Master Pages

I mentioned that one advantage of using Master Pages is that you can create child Master Pages by nesting one Master Page inside of another to provide different submenus or other section differences among various parts of your site. In the Exercise 12-3, we will create a nested Master Page with a section menu in the side column and a place for advertisements to a new master created from `fewd.master`.

We will use the `text.aspx` page created in the first part of this chapter to create our new nested Master Page. Replacing the `sidecol` content in `test.aspx` with a menu showing the chapters with exercise files to download will give us a section menu. A smaller section below the new section menu will be added to hold advertisements or other content. The page is shown in Figure 12-12.



Figure 12-12. In Exercise 12-3, we will create this nested master by starting with a page that is attached to an existing Master Page.

Exercise 12-3. Creating a Nested Master Page

In this exercise, you will create and apply a child Master Page.

1. Begin this exercise by replacing the content of your `sidecol` with the following code:

```
<h3>Exercise Code</h3>
<p>The code used in this book is available for the following chapters:</p>
<ul>
  <li>Chapter 3</li>
  <li>Chapter 4</li>
  <li>Chapter 5</li>
  <li>Chapter 6</li>
  <li>Chapter 7</li>
  <li>Chapter 8</li>
  <li>Chapter 9</li>
  <li>Chapter 10</li>
  <li>Chapter 12</li>
  <li>Chapter 13</li>
  <li>Appendix</li>
</ul>
<p>&nbsp;</p>
```

2. Now that you have the content that will be locked in your new nested Master Page, right-click anywhere in a `ContentPlaceHolder` region in Design view, and select `Manage Microsoft ASP.NET Content Regions`, as shown in Figure 12-13.

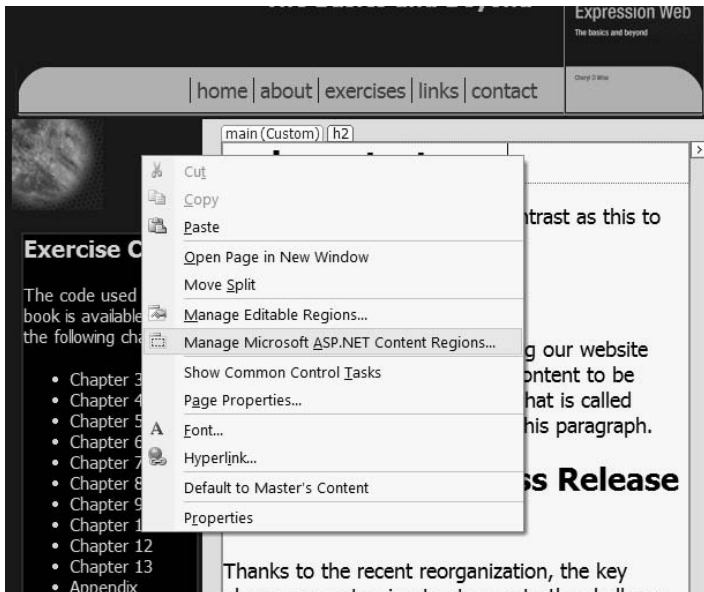


Figure 12-13. You create a nested Master Page using the right-click option `Manage Microsoft ASP.NET Content Regions`.

- When you choose this option, you will be warned that the current page must be saved as a Master Page before you can manage editable regions, as shown in Figure 12-14. Click Yes to continue.

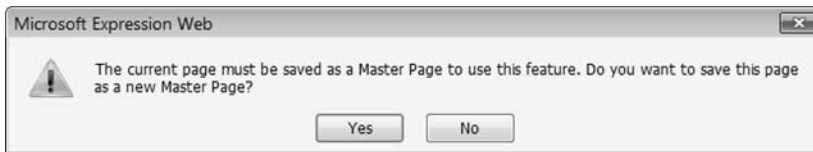


Figure 12-14. *Expression Web warns you before saving as a master.*

- Make sure you save your Master Page with a meaningful name, as shown in Figure 12-15. I suggest using the section name if you will be using the nested master on just one section of your site.

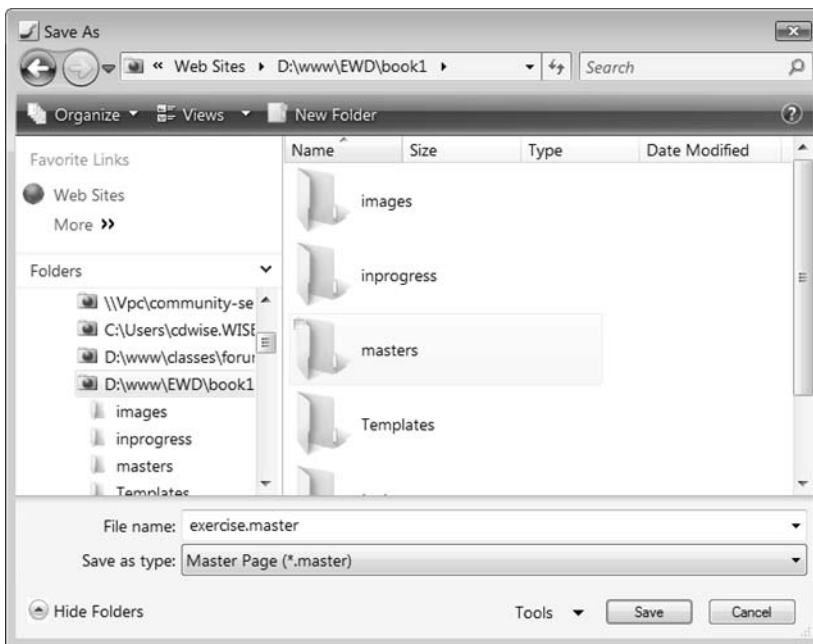


Figure 12-15. *Naming the nested master after the name of the section it holds helps you maintain your site.*

- Unlike when the first Master Page was created, no content placeholders are added to your nested Master Page. Expression warns you that there are no ContentPlaceholder regions in your new master, as shown in Figure 12-16. Save it anyway, since you cannot add a ContentPlaceholder until the page is saved as a Master Page.

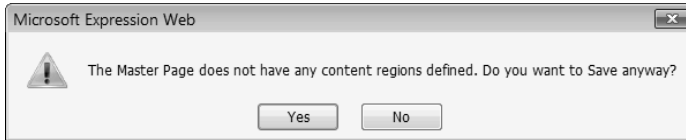


Figure 12-16. You must save the page before you can add ContentPlaceHolders.

6. The Manage Content Regions dialog box will pop up as soon as you save the page. If you add a ContentPlaceholder at this time, the content placeholder will be placed at the same location as your cursor. Instead, close the dialog box.
7. To wrap a selected area in a new ContentPlaceholder, select the part of the page where you want to be able to add content, and launch the Manage Microsoft ASP.NET Content Regions from the right-click menu; see Figure 12-13. Add one ContentPlaceholder to the left menu below the chapter list around the `<p> </p>` called `adverts` and one around the entire main content section named `mainsub`. Save your page once you have added your ContentPlaceHolders.
8. You can now use `exercise.master` to create pages for the exercise section the same way the `text.aspx` page was created, by using **New** ► **Create from Master Page**, as shown in Figure 12-17.

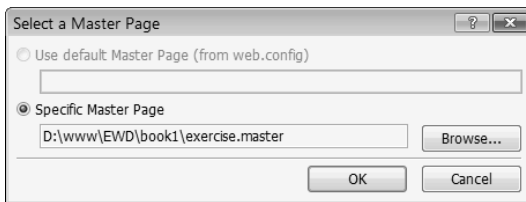


Figure 12-17. Creating a page from a nested master is the same as creating one from any master.

Once you have created a web page from your child Master Page, don't forget to save your new page and then compare it with the `text.aspx` page you created in Exercise 12-2. Don't forget to add your `asp:Content` areas.

Compare the pages in Figure 12-18. On the left is `test.aspx`, which was created from `fewd.master`, and on the right is a page named `chapter3.aspx`, which was created from `exercise.master` to be the default page for Chapter 3's downloads. In the `text.aspx` page, the entire left column is a ContentPlaceholder. In the `chapter3.aspx` page's chapter list, notice the forbidden symbol that indicates the text is outside of the ContentPlaceholder section of the nested Master Page.

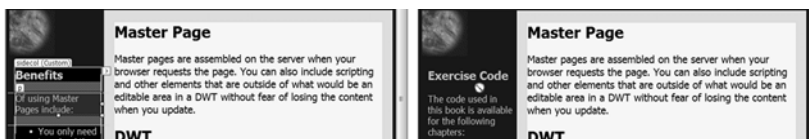


Figure 12-18. The page on the left uses the main master, while the right one uses the nested master.

To illustrate how updating the primary master (`fewd.aspx`) will update the nested Master Page (`exercise.master`) and all pages attached to both Master Pages, I have edited the footer to add links to the site privacy policy and to a site map, as shown in Figure 12-19.



Figure 12-19. Update the `fewd.master` page by adding new links.

There will be no dialog boxes or other prompts to inform you of the change, but when you look at `chapter3.aspx`, you will see that all the links you added to the `fewd.master` page are there, as shown in Figure 12-20.



Figure 12-20. `Chapter3.aspx` immediately reflects the changes to `fewd.master`.

Notice that since `chapter3.aspx` has not had the default content from `exercise.master` replaced with custom content, the only code you see is the reference to `exercise.master`. If we made a change to `exercise.master`, the changes would only appear on the pages that use `exercise.master` and not those made directly from `fewd.master`.

Summary

This chapter explored the unique ability to nest Master Pages to create subsections of your page. With Master Pages, you can use ASP.NET controls to add new content and additional ASP.NET controls. As you have seen, Master Pages and DWTs perform similar functions. The primary difference is where the resulting page is created. With DWTs, the page is created inside Expression Web. DWTs depend on the editor for updates, and all changed pages must be uploaded to the web server. Master Pages are editor independent, and updates are effective as soon as the changed Master Page is uploaded. In addition, you can use a nested or child Master Page to independently manage subsections of pages while still enjoying the benefit of a sitewide parent master. For these reasons, if you are hosted on a web server that supports ASP.NET 2.0, Master Pages are a better option for managing your site than DWTs.