Walkthrough: Using Nested Master Pages in ASP.NET

<http://msdn.microsoft.com/en-us/library/bb547109(v=vs.100).aspx>

**.NET Framework 4**

[Other Versions](javascript:;)

16 out of 26 rated this helpful - [Rate this topic](http://msdn.microsoft.com/en-us/library/bb547109(v=vs.100).aspx#feedback)

This walkthrough describes how to create a master page that is nested inside another master page. Master pages function as templates that specify how pages will look.

You can nest master pages so that each page can have a flexible layout but also retain a consistent look on a Web site. For example, you could create a parent master page that has a company banner at the top and site navigation controls in a side column. You could then create a child master page for a specific department or product that uses the parent master page. It could also act as a master page for all other related department or product pages. In this manner, each product line or department would have a consistent look, and all pages would have a consistent overall look by using the parent master page. For more information about master pages, see [ASP.NET Master Pages](http://msdn.microsoft.com/en-us/library/wtxbf3hh(v=vs.100).aspx).

This walkthrough illustrates the following tasks:

* Creating a master page.
* Creating a master page that is nested inside another master page.
* Creating an ASP.NET page with content that you want to display in the master page.

[Prerequisites](javascript:void(0))

To complete this walkthrough, you will need the following:

* Visual Studio 2008 or Microsoft Visual Web Developer Express. For download information, see the [Visual Studio Developer Center](http://go.microsoft.com/fwlink?linkid=27895) Web site.
* The .NET Framework version 3.5.
* Optionally, a .jpg, .gif, or other graphics file that you can use as a banner on the master page. It is recommended that the banner be no more than 780 pixels wide and 150 pixels in height. However, displaying a logo is optional and the graphic's exact size is not critical to completing the walkthrough.

[Creating a Web Site](javascript:void(0))

If you have already created a Web site in Visual Web Developer (for example, by following the steps in [Walkthrough: Creating a Basic Web Page in Visual Studio](http://msdn.microsoft.com/en-us/library/k4cbh4dh(v=vs.100).aspx)), you can use that Web site and go to the next section, "Creating the Master Page." Otherwise, create a new Web site and page.

This walkthrough uses a Web site project. You could use a Web application project instead. For information about the difference between these Web project types, see [Web Application Projects versus Web Site Projects](http://msdn.microsoft.com/en-us/library/dd547590(v=vs.100).aspx).

To create a file system Web site

1. Open Visual Web Developer.
2. In the **File** menu, click **New** **Web Site**.

The **New Web Site** dialog box is displayed.

1. Under **Visual Studio installed templates**, click **ASP.NET Web Site**.
2. In the **Location** box, type the name of the folder where you want to keep the pages of the Web site.

For example, you could type the following: **C:\Tasks**

1. In the **Language** list, click the programming language that you prefer to work with.
2. Click **OK**.

Visual Web Developer creates the folder and a new page named Default.aspx.

[Creating a Parent Master Page](javascript:void(0))

In this section, you create a parent master page. This page contains a banner and navigation controls that can be used throughout the site. Later, you will create another master page that will be used inside this parent master page. Child master pages can provide various layouts for pages while retaining the look that is established by the parent master page.

To create the master page

1. In **Solution Explorer**, right-click the name of the Web site, and then click **Add** **New Item**.

The **Add** **New Item** dialog box is displayed.

1. Under **Visual Studio installed templates**, click**Master Page**.
2. In the **Name** box, type **ParentMaster**.
3. Clear the **Place code in separate file** check box.
4. In the **Language** list, click the programming language that you prefer to work with.
5. Click **Add**.

The new master page is opened in **Source** view.

[Adding Elements to the Parent Master Page](javascript:void(0))

At the top of the master page is an [@ Master](http://msdn.microsoft.com/en-us/library/ms228176(v=vs.100).aspx) declaration instead of the [@ Page](http://msdn.microsoft.com/en-us/library/ydy4x04a(v=vs.100).aspx) declaration that is ordinarily found at the top of ASP.NET pages. The body of the page contains a [ContentPlaceHolder](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.contentplaceholder(v=vs.100).aspx) control, which is the area of the master page where replaceable content will be merged with content pages at run time. You will work more with content placeholders later in this walkthrough.

Adding Graphics to the Parent Master Page

Before you can incorporate graphics into the parent master page, you have to add the graphic files to the Web site.

Create a graphic that is 780 pixels wide and 150 pixels in height for the banner and a graphic that is 780 pixels wide and 50 pixels in height for the footer. The graphics are used to show how graphics on the parent master page will appear in a nested master page. If you do not have these graphics, you can create them in Microsoft Paint or another graphics program.

To add an existing graphic file to the Web site

1. In **Solution Explorer**, right-click the name of the Web site, and then click **Add** **Existing Item**.

The **Add Existing Item** dialog box is displayed.

1. Select the graphic files that you created.
2. Click **Add**.

On the parent master page, you add two simple graphics, one for a banner and the other for a footer. These graphics let you see that the parent master page is being used in the nested master pages that you will create later in this walkthrough.

To add banner and footer graphics to the parent master page

1. In **Source** view, immediately after the opening **<form>** tag, add a **div** element that contains an **img** element for the banner graphic.

For example, if you created a graphic named Banner.gif, the markup to add this graphic would resemble this:

<div id="banner">

<img src="banner.gif" alt="banner graphic" />

</div>

1. Immediately before the closing **</form>** tag, add a **div** element that contains an **img** element for the footer graphic.

For example, if you created a graphic named Footer.gif, the markup to add this graphic would resemble this:

<div id="banner">

<img src="footer.gif" alt="footer graphic" />

</div>

1. Save the master page.

Notice that you have not added anything inside the [ContentPlaceHolder](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.contentplaceholder(v=vs.100).aspx) control. The next set of master pages that you create will be inside the[ContentPlaceHolder](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.contentplaceholder(v=vs.100).aspx) control.

[Creating a Child Master Page](javascript:void(0))

To nest one master page inside another, you must create another master page. The new master page will be inside the content placeholder of the parent master page. The child master page lets you arrange the page in different ways while maintaining a consistent appearance that is established by the parent master page.

To create the child master page

1. In **Solution Explorer**, right-click the name of the Web site, and then click **Add** **New Item**.

The **Add New Item** dialog box is displayed.

1. Under **Visual Studio installed templates**, click **Master Page**.
2. In the **Name** box, type **ChildMaster**.
3. Clear the **Place code in separate file** check box.
4. Select the **Select master page** check box.
5. In the **Language** list, click the programming language that you prefer to work with.
6. Click **Add**.

The **Select a Master Page** dialog box is displayed.

1. Select the parent master page that you created earlier in this walkthrough.
2. Click **OK**.

The new master page is opened in **Source** view.

Notice that the [@ Master](http://msdn.microsoft.com/en-us/library/ms228176(v=vs.100).aspx) declaration at the top of the new master page indicates that it references a master page.

Adding ContentPlaceHolder Controls to the Child Master Page

Because the child master page has another master page associated with it, it contains two [Content](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.content(v=vs.100).aspx) controls. The first content placeholder can be used to add information to the page that would ordinarily appear in the **head** element, such as a **script** element. Inside the second [Content](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.content(v=vs.100).aspx) control you can add a[ContentPlaceHolder](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.contentplaceholder(v=vs.100).aspx) control. This lets an ASP.NET page that uses the child master page provide the content for the page. You can add other page elements inside the [Content](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.content(v=vs.100).aspx) control. The child master page can contain additional page elements to provide a consistent look to pages that use the child master page.

To add ContentPlaceholder controls to the child master page

1. Open or switch to the child master page
2. Switch to **Source** view.
3. Add the following markup inside the second [Content](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.content(v=vs.100).aspx) control section in order to create [ContentPlaceHolder](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.contentplaceholder(v=vs.100).aspx) controls:
4. <div id="2col">
5. <asp:ContentPlaceHolder ID="leftcolumn" runat="server">
6. </asp:ContentPlaceHolder>
7. <asp:ContentPlaceHolder ID="rightcolumn" runat="server">
8. </asp:ContentPlaceHolder>
9. </div>
10. Save the file.

The child master page now has [ContentPlaceHolder](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.contentplaceholder(v=vs.100).aspx) controls that contain markup from an ASP.NET page that uses the child master page.

[Creating a Content Page That Uses the Child Master Page](javascript:void(0))

In the previous steps, you created a master page that is nested inside another master page. The resulting child master page lets you use the UI elements of the parent master page. It also provides additional UI elements in the second master. In this walkthrough, you added graphics to the parent master page and placeholders in the second master page. To see the nested master pages in action, you have to create an ASP.NET Web page that uses the child master page. The new page that you create that uses the child master page will automatically contain a [Content](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.content(v=vs.100).aspx) control for each [ContentPlaceHolder](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.contentplaceholder(v=vs.100).aspx) control that you created in the child master page.

To create a page that uses the child master page

1. In **Solution Explorer**, right-click the name of the Web site, and then click **Add** **New Item**.

The **Add New Item** dialog box is displayed.

1. Under **Visual Studio installed templates**, click **Web Form**.
2. In the **Name** box, type **Tasks**.
3. Select the **Place code in separate file** check box.
4. Select the **Select master page** check box.
5. In the **Language** list, click the programming language that you prefer to work with.
6. Click **Add**.
7. In the **Select a Master Page** dialog box, select the child master page that you created earlier in this walkthrough.
8. Click **OK**.

The new content page is opened in **Source** view.

1. Change the ID of the first [Content](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.content(v=vs.100).aspx) control to **leftcolumn** and change the ID of the second [Content](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.content(v=vs.100).aspx) control to **rightcolumn**.
2. Add text or page elements to the left or right column [Content](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.content(v=vs.100).aspx) controls.
3. Press CTRL+F5 to run the Web site.

Notice that the Tasks.aspx page displays a combination of all the elements that you created in this walkthrough. This includes the graphics on the parent master page, the two-column layout on the child master page, and text and controls you added to the Tasks.aspx page.

[Next Steps](javascript:void(0))

Now that you have seen how nested master pages work, you can create other child master pages that use the parent master page. The child master page that you created in this walkthrough created a two-column layout. Another child master page that you create for the parent master page could create a single-column layout or some other layout.

[See Also](javascript:void(0))

Tasks

[Walkthrough: Creating and Modifying a CSS File](http://msdn.microsoft.com/en-us/library/bb398932(v=vs.100).aspx)