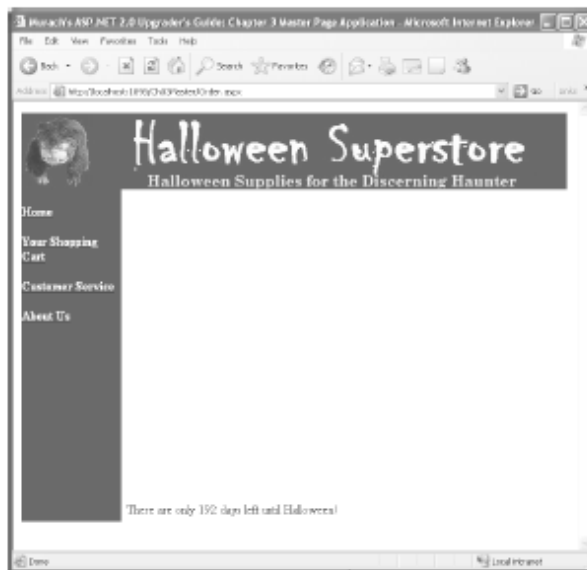


Master Pages:

A master page is a page that provides a framework within which the content from other pages can be displayed. Master pages make it easy to include banners, navigation menus, and other elements on all of the pages in an application. In the topics that follow, you'll learn how to create master pages in your ASP.NET applications.

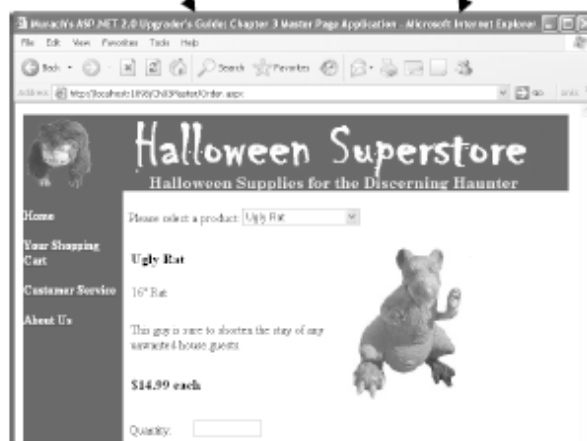
Master page (MasterPage.master)



Content page (Order.aspx)



Rendered page



Description

- A *master page* provides a framework in which the content of each page on a web site is presented. Master pages make it easy to create pages that have a consistent look.
- The pages that provide the content that's displayed in a master page are called content

pages.

- The content of each content page is displayed in the master page's content placeholder.

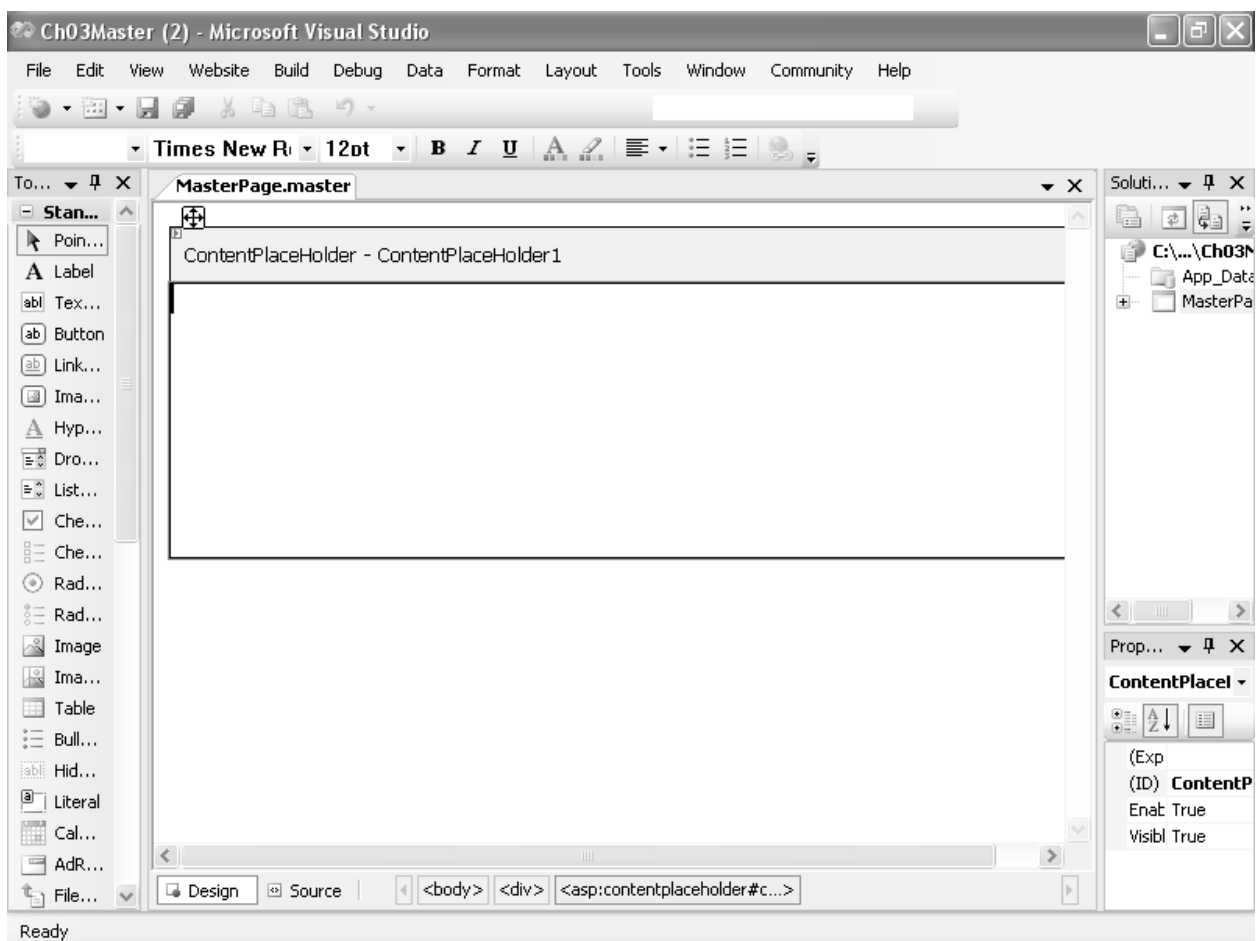
How to create a master page

create a master page by using the Website_Add New Item command. Master Page is listed as one of the templates in the Add New Item dialog box. Select this template, select Visual C# as the language, and click Add to create the master page. The default name for a master page is MasterPage.master.

The master page created from the template includes a ContentPlaceHolder control that will contain the content page, but nothing else.

You can develop the master page by adding additional elements outside of the ContentPlaceHolder control.

A new master page in Design view



Description

- To add a master page to a project, choose the Website_Add New Item command. Then, in the Add New Item dialog box, select Master Page from the list of templates, specify the name of the master page you want to create in the Name text box (the default is Master Page.master), and select the programming language. Then, click Add.
- The content placeholder appears as a control in the Web Forms Designer. Although you can change the position of the content placeholder, you can't edit its contents from the master page. Instead, you add content to the master page by creating content pages as described later in this chapter.
- Any elements you add to the master page outside of the content placeholder will appear on every content page that uses the master page.
- Although most master pages have just one content placeholder, you can create more than one content placeholder if you need to. In that case, each placeholder displays a portion of the content of each content page.
- An application can have more than one master page, and each content page specifies which master page should be used to display the content page.
- The aspx file for a master page uses the extension .master. The code-behind file uses .master.cs.

The aspx code for a new master page

```
<%@ Master Language="C#" AutoEventWireup="true"
CodeFile="MasterPage.master.cs" Inherits="MasterPage_master" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/
xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
<title>Untitled Page</title>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:contentplaceholder id="ContentPlaceholder1" runat="server">
</asp:contentplaceholder>
</div>
</form>
</body>
</html>
```

The aspx code for a new master page

When you create a master page using the Master Page template. As you can see, this code is similar to the aspx code generated for a regular ASP.NET web page, with two important differences.

First, instead of a Page directive, the code begins with a Master directive. This indicates that the file contains a master page rather than a regular ASP.NET page. Second, the Div element that normally contains the content for the page now contains a ContentPlaceHolder control.

Notice that the master page file is itself a well-formed HTML document with html, head, and body elements. The body element includes a form element, which in turn contains the ContentPlaceHolder control. Any elements you add to the master page should appear within the form element, but outside of the ContentPlaceHolder control.

Attributes of the Master page directive

Attribute	Description
Language	Specifies the language used for any code required by the page.
AutoEventWireup	Specifies whether event-handling methods should be automatically wired.
CodeFile	Specifies the name of the code-behind file.
Inherits	Specifies the name of the page class defined in the code-behind file.

Attributes of the ContentPlaceHolder control

Attribute	Description
ID	Specifies the name of the content placeholder.
Runat	Specifies that the control is a server-side control.