

# Chapter 6

## Customizing and Managing Your Site's Appearance

PRENTICE  
HALL

### core

# INTERNET

APPLICATION DEVELOPMENT  
WITH **ASP.NET 2.0**



▼ Get up to speed quickly with numerous real-world walkthrough exercises, code listings, in-depth examples, and snippets

▼ Learn valuable design principles and best practices of ASP.NET Web application development

▼ Gain in-depth knowledge of many key ASP.NET 2.0 topics, including controls, data binding, security, Web services, and a sneak peek at ASP.NET AJAX

RANDY CONNOLLY

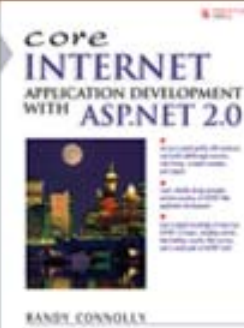
---

# Overview

- This chapter covers how to customize and manage your site's appearance in ASP.NET.
- It covers:
  - Server control appearance properties
  - Using CSS with ASP.NET
  - Skins and Themes
  - Master Pages
  - User Controls

## 2 Customizing Site Appearance

COPYRIGHT © 2007 RANDY CONNOLLY



CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

Prentice Hall, 2007

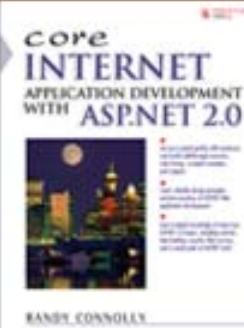
[www.randyconnolly.com/core](http://www.randyconnolly.com/core)

## Common Appearance Properties

- See Table 6.1 on page 312.
- These properties can be set declaratively or programmatically.
- These properties are then rendered in the browser as inline CSS styles.

### 3 Customizing Site Appearance

COPYRIGHT © 2007 RANDY CONNOLLY



CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

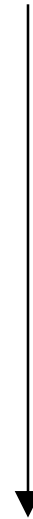
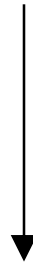
Prentice Hall, 2007

[www.randyconnolly.com/core](http://www.randyconnolly.com/core)

# Appearance Properties

```
<asp:Label id="labTest" runat="server" Text="Hello"  
  ForeColor="#CC33CC" BackColor="Blue"/>
```

```
labTest.ForeColor = Color.FromName("#CC33CC");  
labTest.BackColor = Color.Blue;
```



```
<span id="labTest" style="color: #CC33CC; background-color:blue;">  
Hello  
</span>
```

CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

Prentice Hall, 2007  
[www.randyconnolly.com/core](http://www.randyconnolly.com/core)

Core  
INTERNET  
APPLICATION DEVELOPMENT  
WITH ASP.NET 2.0



RANDY CONNOLLY

# CSS and ASP.NET

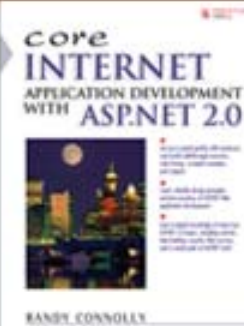
- Although the appearance properties are very useful for customizing the appearance of your Web output, they do not contain the full formatting power of Cascading Style Sheets.
- Fortunately, you can also customize the appearance of Web server controls using CSS via the `CssClass` property.

```
<asp:Label id="labTest" runat="server" Text="Hello" CssClass="quote" />
```

```
<style>  
  .quote { color: green; background-color: red; }  
</style>
```

## 5 Customizing Site Appearance

COPYRIGHT © 2007 RANDY CONNOLLY



CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

Prentice Hall, 2007

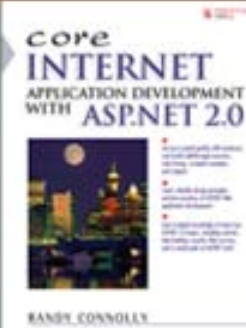
[www.randyconnolly.com/core](http://www.randyconnolly.com/core)

# Why CSS?

- There are numerous advantages of using CSS:
  - Greater control over textual effects and page layout.
  - **Separates document content (HTML) from its appearance (CSS).**
  - Reducing duplication of style commands.
  - Increase ease in web site maintenance.
- The disadvantages of CSS is:
  - Harder to learn than straight HTML
  - Some unevenness with browser support
    - IE 6 and earlier in particular has several CSS bugs

## 6 Customizing Site Appearance

COPYRIGHT © 2007 RANDY CONNOLLY



CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

Prentice Hall, 2007

[www.randyconnolly.com/core](http://www.randyconnolly.com/core)

# ASP.NET and CSS

- Unfortunately, many ASP.NET authors do not fully take advantage of CSS, and instead litter their web server controls with numerous appearance property settings (e.g., BackColor, BorderColor, etc).
- While it is true that these properties are rendered as inline CSS styles, the use of these properties still eliminates the principal benefit of CSS:
  - the ability to centralize all appearance information for the web page or web site into one location, namely, an external CSS file.
- As well, since the appearance properties are rendered as inline CSS, this will increase the size as well as the download time of the rendered page.

## 7 Customizing Site Appearance

COPYRIGHT © 2007 RANDY CONNOLLY



CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

Prentice Hall, 2007

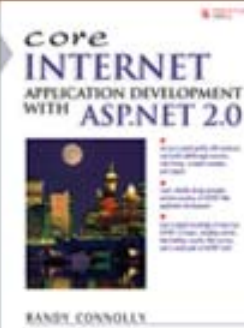
[www.randyconnolly.com/core](http://www.randyconnolly.com/core)

# CSS and ASP.NET

- By limiting the use of appearance properties for web server controls within your web forms, and using instead an external CSS file to contain all the site's styling, your web form's markup will become simpler and easier to modify and maintain.
- For instance, rather than setting the Font property declaratively for a dozen web server controls in a page to the identical value, it would make much more sense to do so via a single CSS rule.
  - And if this CSS rule is contained in an external CSS file, it could be used throughout the site (that is, in multiple web forms), reducing the overall amount of markup in the site.

## 8 Customizing Site Appearance

COPYRIGHT © 2007 RANDY CONNOLLY



CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

Prentice Hall, 2007

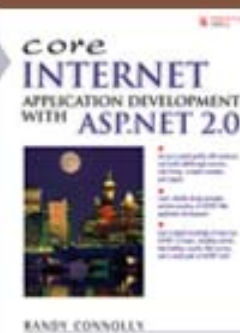
[www.randyconnolly.com/core](http://www.randyconnolly.com/core)

# Themes and Skins

- Themes and skins are an additional mechanism in ASP.NET 2.0 for centralizing the setting the appearance of web server controls on a site-wide basis.
- Like CSS, ASP.NET themes allow you to separate web server control styling from the pages themselves.
  - They have the additional benefit of having a complete object model that can be manipulated programmatically.
- Themes still allow you to use CSS for the majority of your visual formatting.

## 9 Customizing Site Appearance

COPYRIGHT © 2007 RANDY CONNOLLY



CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

Prentice Hall, 2007

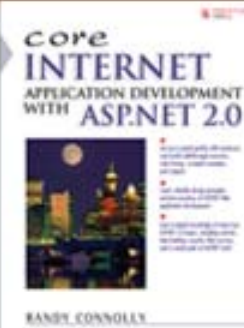
[www.randyconnolly.com/core](http://www.randyconnolly.com/core)

# Themes are Programmatic

- By using CSS and themes, you can dramatically alter the appearance of your web site with a single line of programming.

## 10 Customizing Site Appearance

COPYRIGHT © 2007 RANDY CONNOLLY



CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

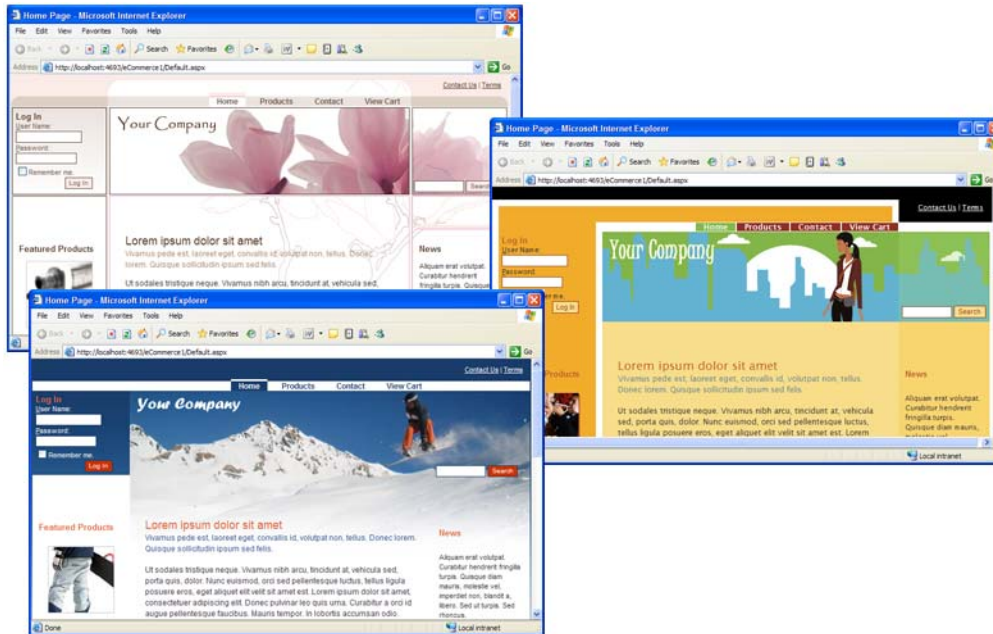
Prentice Hall, 2007

[www.randyconnolly.com/core](http://www.randyconnolly.com/core)

# Themes are Programmatic

11 Chapter title here

COPYRIGHT © 2007 RANDY CONNOLLY



CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

Prentice Hall, 2007  
[www.randyconnolly.com/core](http://www.randyconnolly.com/core)

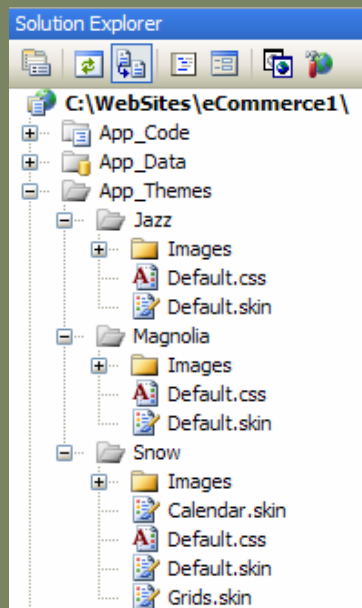
Core  
INTERNET  
APPLICATION DEVELOPMENT  
WITH ASP.NET 2.0



RANDY CONNOLLY

# Themes and Skins

- An ASP.NET web application can define multiple **themes**.
  - Each theme resides in its own folder within the `App_Themes` folder in the root of your application.
- Within each theme folder, there will be one or more **skin** files, as well as optional subfolders and CSS and image files



## 12 Customizing Site Appearance

COPYRIGHT © 2007 RANDY CONNOLLY

Core  
INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0



RANDY CONNOLLY

CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

Prentice Hall, 2007

[www.randyconnolly.com/core](http://www.randyconnolly.com/core)

# Skin

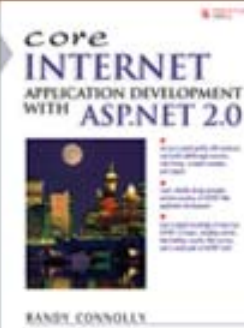
- A skin describes the appearance of one or more control types.
- For example, a skin file might look like the following.

```
<asp:Label runat="server" ForeColor="Blue"  
    Font-Size="10pt" Font-Name="Verdana" />
```

- Notice that a skin simply contains a control definition *without* the `id` attribute.
- A given skin file can contain multiple control definitions.

## 13 Customizing Site Appearance

COPYRIGHT © 2007 RANDY CONNOLLY



CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

Prentice Hall, 2007

[www.randyconnolly.com/core](http://www.randyconnolly.com/core)

# Theme

- Themes reside in separate folders within the `App_Themes` folder within your site.
- Once a theme has been created (that is, once you've created one or more skin files), you can apply a theme to a page, via the `Page` directive:

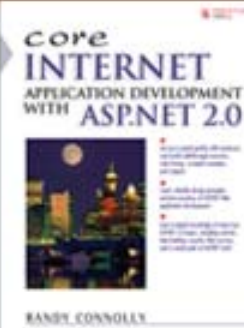
```
<%@ Page ... Theme="Cool" %>
```

- Or programmatically by setting the page's `Theme` property in the `PreInit` event handler for the page.

```
protected void Page_PreInit(object o, EventArgs e)
{
    //set theme for this page
    this.Theme = "Cool";
}
```

## 14 Customizing Site Appearance

COPYRIGHT © 2007 RANDY CONNOLLY



CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

Prentice Hall, 2007

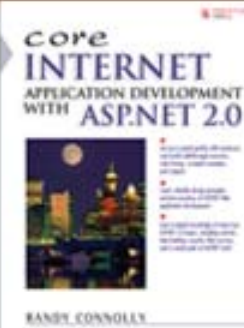
[www.randyconnolly.com/core](http://www.randyconnolly.com/core)

# Master Pages

- Allows the developer to define the structural layout for multiple web forms in a separate file and then apply this layout across multiple web forms.
- You can thus move common layout elements, such as logos, navigation systems, search boxes, log-in areas, and footers, out of all the individual pages and into a single master page.
- They allow the developer to create a consistent page structure or layout without duplicating code or markup.

## 15 Customizing Site Appearance

COPYRIGHT © 2007 RANDY CONNOLLY



CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

Prentice Hall, 2007

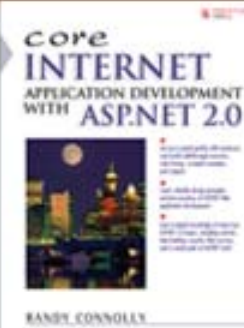
[www.randyconnolly.com/core](http://www.randyconnolly.com/core)

# Master Pages

- Master pages are created in much the same way as any other web form.
- They contain markup, server controls, and can have a code-behind class that responds to all the usual page lifecycle events.
- However, they do have their own unique extension (.master) as well as a different directive (Master) at the top of the page.
- As well (and most importantly) they also contain one or more ContentPlaceHolder controls.

## 16 Customizing Site Appearance

COPYRIGHT © 2007 RANDY CONNOLLY



CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

Prentice Hall, 2007

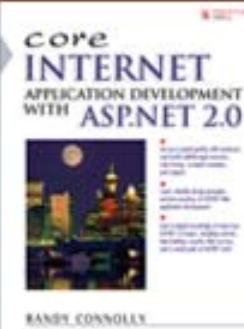
[www.randyconnolly.com/core](http://www.randyconnolly.com/core)

# ContentPlaceHolder

- The `ContentPlaceHolder` control defines a region of the master page which will be replaced by content from the page that is using this master page.
- That is, each web form that uses the master page only needs to define the content unique to it within the `Content` controls that correspond to the `ContentPlaceHolder` controls in the master page.

## 17 Customizing Site Appearance

COPYRIGHT © 2007 RANDY CONNOLLY



CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

Prentice Hall, 2007

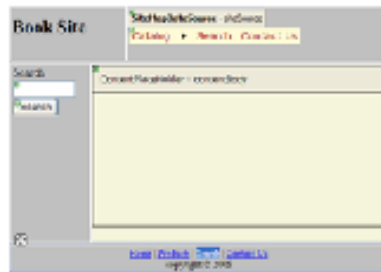
[www.randyconnolly.com/core](http://www.randyconnolly.com/core)

# ContentPlaceHolder

18 Chapter title here

COPYRIGHT © 2007 RANDY CONNOLLY

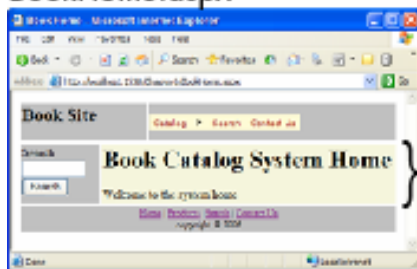
## Books.Master



## ContentPlaceHolder

```
<%@ Master ... %>
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">...</head>
<body>
  <form id="form1" runat="server">
    ...
    <asp:ContentPlaceHolder id="contentBody"
      runat="server"/>
    ...
  </form>
</body>
</html>
```

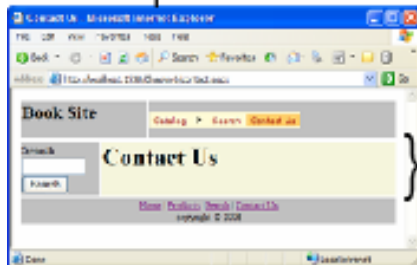
## BookHome.aspx



## Content

```
<%@ Page ... MasterPageFile="-/Books.master" %>
<asp:Content ID="Content1" Runat="Server"
  ContentPlaceHolderID="contentBody" >
  <h1>Book Catalog System Home</h1>
  <p>Welcome to the system home</p>
</asp:Content>
```

## Contact.aspx



## Content

```
<%@ Page ... MasterPageFile="-/Books.master" %>
<asp:Content ID="Content1" Runat="Server"
  ContentPlaceHolderID="contentBody" >
  <h1>Contact Us</h1>
</asp:Content>
```

CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

Prentice Hall, 2007  
[www.randyconnolly.com/core](http://www.randyconnolly.com/core)

Core  
INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0



RANDY CONNOLLY

# ContentPlaceHolder

- A master page significantly simplifies the markup for pages that use it.
- The master page contains the complete XHTML document structure.
  - That is, the html, head, and body elements are contained only within the master page.
- The .aspx files that use the master page only need define the content that will be inserted into the placeholders in the master page.
  - In fact, these .aspx files can *only* contain content within Content controls.
- Master pages are transparent to the user (that is, the user is unaware of their existence), since ASP.NET merges the content of the master page with the content of the requested page.

## 19 Customizing Site Appearance

COPYRIGHT © 2007 RANDY CONNOLLY

Core  
INTERNET  
APPLICATION DEVELOPMENT  
WITH ASP.NET 2.0



RANDY CONNOLLY

CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

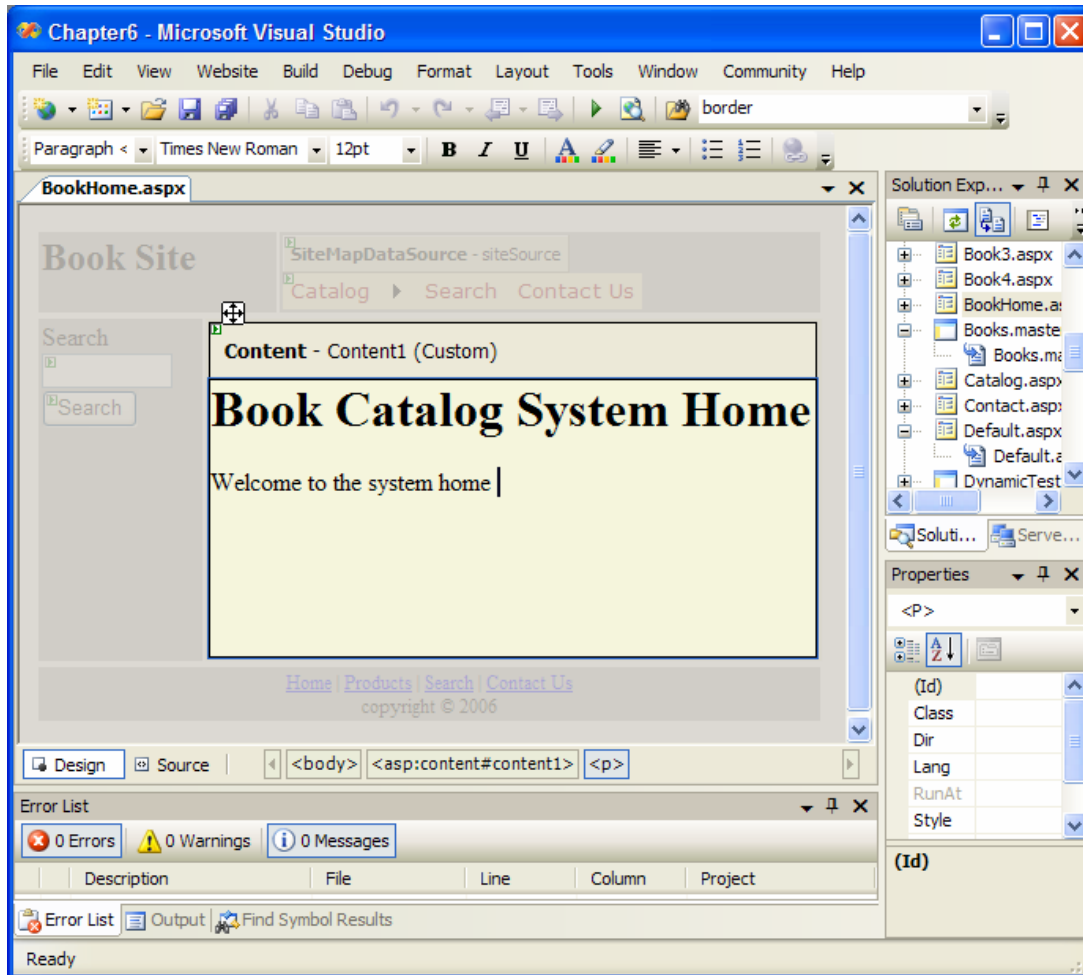
Prentice Hall, 2007

[www.randyconnolly.com/core](http://www.randyconnolly.com/core)

# Visual Studio + Master Pages

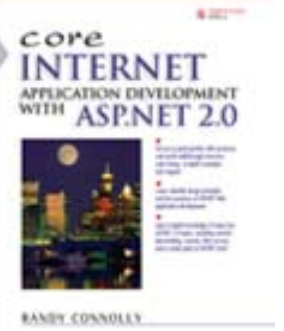
20 Chapter title here

COPYRIGHT © 2007 RANDY CONNOLLY



CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

Prentice Hall, 2007  
[www.randyconnolly.com/core](http://www.randyconnolly.com/core)

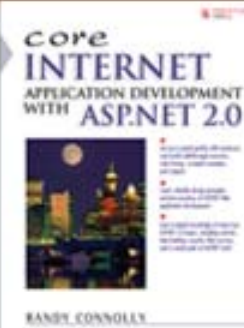


# Nested Master Pages

- Master pages can be nested so that one master page contains another master page as its content.
- This can be particularly useful for Web sites that are part of a larger system of sites.

21 Customizing Site Appearance

COPYRIGHT © 2007 RANDY CONNOLLY



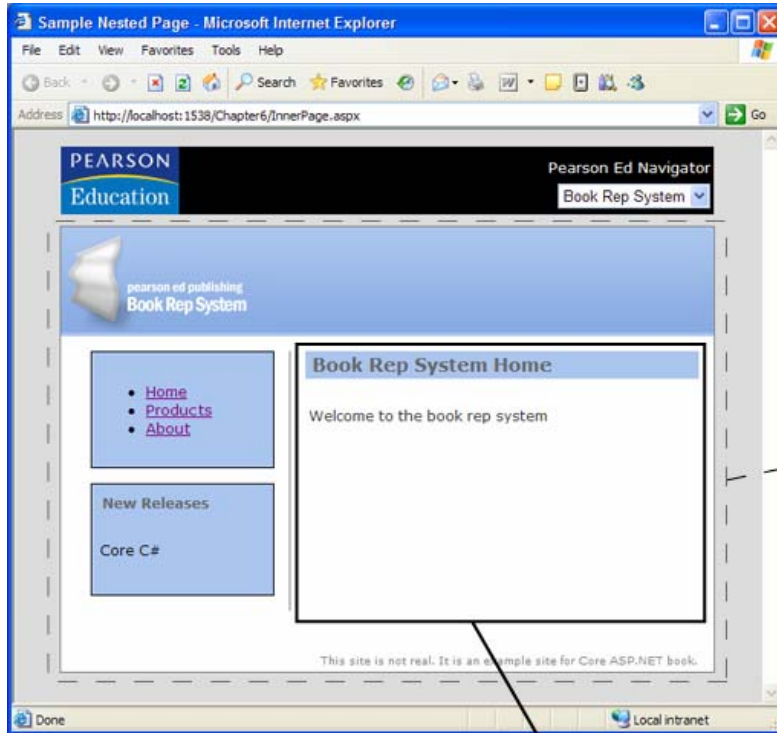
CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

Prentice Hall, 2007  
[www.randyconnolly.com/core](http://www.randyconnolly.com/core)

# Nested Master Pages

22 Chapter title here

COPYRIGHT © 2007 RANDY CONNOLLY



parent master page  
ContentPlaceholder

child master page  
ContentPlaceholder

CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

Prentice Hall, 2007  
www.randyconnolly.com/core

core  
INTERNET  
APPLICATION DEVELOPMENT  
WITH ASP.NET 2.0



RANDY CONNOLLY

# How Master Pages Work

- When a page that uses a master page (i.e., a content page) is requested, ASP.NET merges the content page and master page together.
- It does so by inserting the master page's content at the beginning of the content page's control tree.
  - **This means that the master page content is actually a control that is added to the page.**
  - In fact, the master page control is a subclass of the `UserControl` class.

## 23 Customizing Site Appearance

COPYRIGHT © 2007 RANDY CONNOLLY

Core  
**INTERNET**  
APPLICATION DEVELOPMENT  
WITH **ASP.NET 2.0**



RANDY CONNOLLY

CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

Prentice Hall, 2007

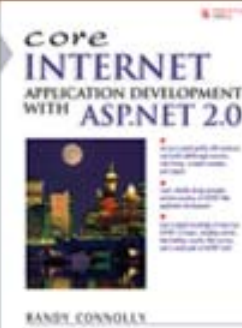
[www.randyconnolly.com/core](http://www.randyconnolly.com/core)

# Master Pages and the Page Lifecycle

- Like any control, master pages have their own sequence of events that are fired as part of the page lifecycle.
- Because it is the root control in the page's control hierarchy, its events are always fired **before** the control events in any content pages.
  - The master page should thus *not* be orchestrating and controlling what happens in the individual content pages it contains.
  - The master page is simply a template page and should control only those server controls it directly contains.

24 Customizing Site Appearance

COPYRIGHT © 2007 RANDY CONNOLLY



CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

Prentice Hall, 2007

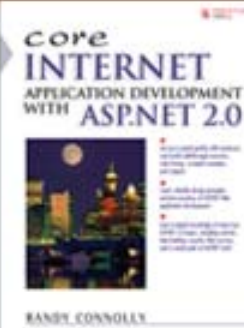
[www.randyconnolly.com/core](http://www.randyconnolly.com/core)

# User Controls

- It still is possible, even when using a master page, for presentation-level (i.e., user interface) duplication to exist in a site.
  - For instance, there might be some user interface element that appears on several but not all content pages.

## 25 Customizing Site Appearance

COPYRIGHT © 2007 RANDY CONNOLLY



CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

Prentice Hall, 2007

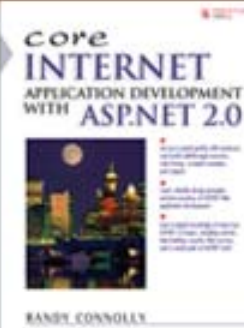
[www.randyconnolly.com/core](http://www.randyconnolly.com/core)

# User Controls

- User controls are the preferred ASP.NET solution to this type of presentation-level duplication.
- They provide a cleaner approach to user interface reuse than copying and pasting or using the server-side includes of classic ASP.
  - User controls in ASP.NET are very simple to create and then use.
  - They follow the same familiar development model as regular Web forms.

## 26 Customizing Site Appearance

COPYRIGHT © 2007 RANDY CONNOLLY



CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

Prentice Hall, 2007

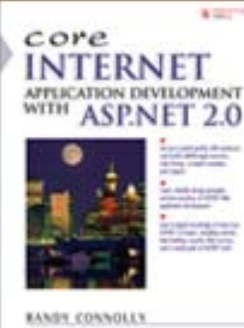
[www.randyconnolly.com/core](http://www.randyconnolly.com/core)

# Creating User Controls

- User controls are created in a manner similar to Web forms.
  - As with Web forms, a user control can contain markup as well as programming logic.
  - Also like Web forms, the programming for a user control can be contained within the same file as the markup, or contained within a separate code-behind file.
- After a user control is defined, it can then be used in Web forms, master pages, or even other user controls.

27 Customizing Site Appearance

COPYRIGHT © 2007 RANDY CONNOLLY



CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

Prentice Hall, 2007

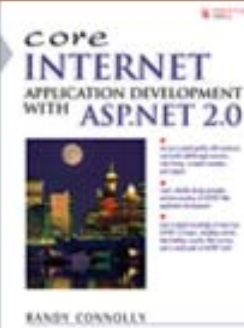
[www.randyconnolly.com/core](http://www.randyconnolly.com/core)

# Creating User Controls

- The markup for a user control is contained in a text file with the `.ascx` extension.
  - This file can contain any necessary programming logic within embedded code declaration block embedded within this `.ascx` user control file.
  - Alternately, the programming code can be contained in a code-behind class for the user control.
  - However, the code-behind class for a user control inherits from the `UserControl` class, rather than the `Page` class.

28 Customizing Site Appearance

COPYRIGHT © 2007 RANDY CONNOLLY



CORE INTERNET  
APPLICATION DEVELOPMENT  
WITH  
ASP.NET 2.0

Prentice Hall, 2007

[www.randyconnolly.com/core](http://www.randyconnolly.com/core)