Web Form Fundamentals

Today You Will Learn

- The Anatomy of an ASP.NET Application
- Introducing Server Controls

The Anatomy of an ASP.NET Application

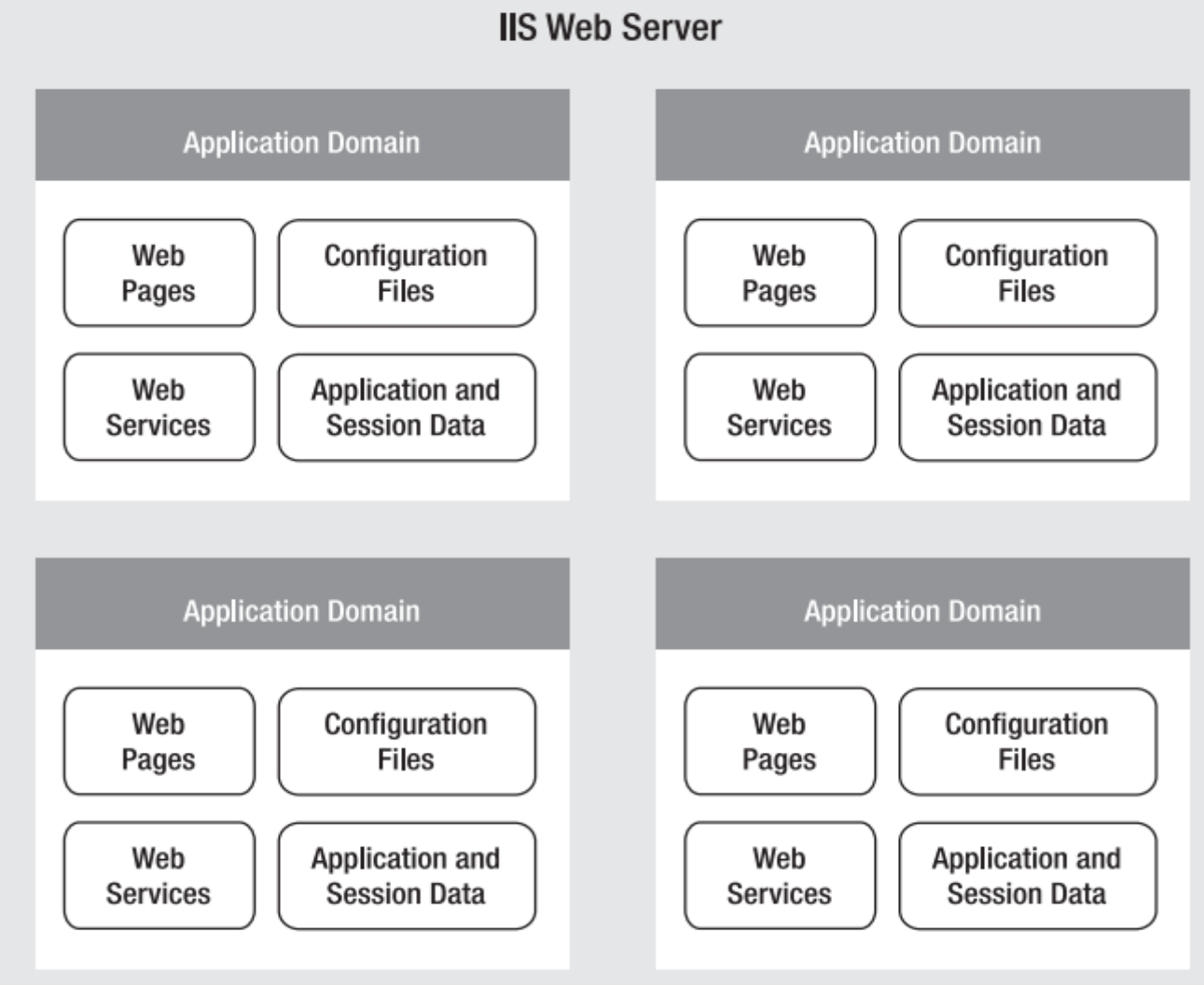
- ASP.NET applications are almost always divided into multiple web pages.
- Web pages from other ASP.NET applications don't share these resources, even if they're on the same web server.
- Every ASP.NET application is executed inside a separate application domain.
- Application domains are isolated areas in memory, and they ensure that even if one web application causes a fatal error, it's unlikely to affect any other application that is currently running on the same computer.

The Anatomy of an ASP.NET Application

- An ASP.NET application is a combination of files, pages, handlers, modules, and executable code that can be invoked from a virtual directory (and, optionally, its subdirectories) on a web server.

- The virtual directory is the basic grouping structure that delimits an application.

The Anatomy of an ASP.NET Application



The Anatomy of an ASP.NET Application

ASP.NET File Types

File Name	Description
Ends with .aspx	These are ASP.NET web pages. They contain the user interface and, optionally, the underlying application code. Users request or navigate directly to one of these pages to start your web application.
Ends with .ascx	These are ASP.NET user controls. User controls are similar to web pages, except that the user can't access these files directly. Instead, they must be hosted inside an ASP.NET web page. User controls allow you to develop a small piece of user interface and reuse it in as many web forms as you want without repetitive code. You'll learn about user controls in Chapter 11.
web.config	This is the configuration file for your ASP.NET application. It includes settings for customizing security, state management, memory management, and much more. You'll get an introduction to the web.config file in this chapter, and you'll explore its settings throughout this book.
global.asax	This is the global application file. You can use this file to define global variables (variables that can be accessed from any web page in the web application) and react to global events (such as when a web application first starts). You'll learn about it later in this chapter.
Ends with .cs	These are code-behind files that contain C# code. They allow you to separate the application logic from the user interface of a web page. We'll introduce the code-behind model in this chapter and use it extensively in this book.

The Anatomy of an ASP.NET Application

ASP.NET Application Directories

- Along with the directories you create, ASP.NET also uses a few specialized subfolders.
- You won't see all these subfolders in a typical application.

Directory	Description
App_Browsers	Contains .browser files that ASP.NET uses to identify the browsers that are using your application and determine their capabilities. Usually, browser information is standardized across the entire web server, and you don't need to use this folder. For more information about ASP.NET's browser support—which is an advanced features that most ordinary web developers can safely ignore—refer to *Pro ASP.NET 4 in VB 2010* (Apress).
App_Code	Contains source code files that are dynamically compiled for use in your application.
App_GlobalResources	Stores global resources that are accessible to every page in the web application. This directory is used in localization scenarios, when you need to have a website in more than one language. Localization isn't covered in this book, although you can refer to *Pro ASP.NET 4 in VB 2010* (Apress) for more information.

The Anatomy of an ASP.NET Application

Directory	Description
App_LocalResources	Serves the same purpose as App_GlobalResources, except these resources are accessible to a specific page only.
App_WebReferences	Stores references to web services, which are remote code routines that a web application can call over a network or the Internet.
App_Data	Stores data, including SQL Server Express database files (as you'll see in Chapter 14). Of course, you're free to store data files in other directories.
App_Themes	Stores the themes that are used to standardize and reuse formatting in your web application. You'll learn about themes in Chapter 12.
Bin	Contains all the compiled .NET components (DLLs) that the ASP.NET web application uses. For example, if you develop a custom component for accessing a database (see Chapter 22), you'll place the component here. ASP.NET will automatically detect the assembly, and any page in the web application will be able to use it. This seamless deployment model is far easier than working with traditional COM components, which must be registered before they can be used (and often reregistered when they change).

Introducing Server Controls

- In old-style web development, programmers had to master the quirks and details of HTML before they could design a dynamic web page.
- Pages had to be carefully tailored to a specific task, and the only way to generate additional content was to generate raw HTML tags.
- ASP.NET solves this problem with a higher-level model of server controls.
- These controls created and configured as an objects.

Introducing Server Controls

- ASP.NET provides two set of server-side controls:
  - **HTML server controls:** These are server-based equivalents for standard HTML elements. So, we work on familiar HTML tags. They are useful when migrating ordinary HTML pages or ASP pages to ASP.NET, because they require the fewest changes.
  - **Web Controls:** They provide a richer object model with a variety of properties for style, formatting details, more events, more closely resemble the controls used for Windows development. Web controls also feature some user interface elements that have no direct HTML equivalent, such as the GridView, Calendar and validation controls.

Introducing Server Controls

View State

- If you enter information in the text box and click the submit button to post the page, the refreshed page will still contain the value you entered in the text box. (In the original example that uses ordinary HTML elements, the value will be cleared every time the page is submitted.) This change occurs because ASP.NET controls automatically retain their state.