In this chapter, you’ll learn some of the core topics that every ASP.NET developer must master. You’ll begin by taking a closer look at the ASP.NET application model, and considering what files and folders belong in a web application. Then you’ll take a closer look at server controls, the basic building block of every web form. Using server controls, you’ll create your first ASP.NET application, by taking a static HTML page and transforming it into a simple, single-page currency converter.

After you have the currency converter example under your belt, you can explore the basics of ASP.NET’s web form model in more detail. You’ll pick up some handy skills along the way, such as the ability to create controls on the fly, navigate from one page to another, and handle special characters in HTML. Finally, you’ll consider the ASP.NET configuration model, which lets you tweak the settings that govern the behavior of your web application.

Understanding the Anatomy of an ASP.NET Application

It’s sometimes difficult to define exactly what a web application is. Unlike a desktop program or smartphone app, ASP.NET websites almost always include multiple web pages. This division means a user can enter an ASP.NET “application” at different points, and follow a link from one page to another part of the website or to another web server. So, does it make sense to treat an entire website as though it were a single application?

In ASP.NET, the answer is yes. Every ASP.NET application shares a common set of resources and configuration settings. Web pages from other ASP.NET applications don’t share these resources, even if they’re on the same web server. Technically speaking, 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 (in this case, that’s the web server). Similarly, application domains restrict a web page in one application from accessing the in-memory information of another application. Each web application is maintained separately and has its own set of cached, application, and session data.

The standard definition of an ASP.NET application describes it as 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. In other words, the virtual directory is the basic grouping structure that delimits an application. Figure 5-1 shows a web server that hosts four separate web applications.
A virtual directory is a directory that’s exposed to other computers on a web server. As you’ll discover in Chapter 26, you deploy your perfected ASP.NET web application by copying it to a virtual directory.

Figure 5-1. ASP.NET applications

ASP.NET File Types

ASP.NET applications can include many types of files. Table 5-1 introduces the essential ingredients.