Overview of ASP.NET MVC Projects

You’ve just experienced building a good-sized MVC application, SportsStore, and picked up a lot of ASP.NET MVC development knowledge along the way. However, this was just one example, and it didn’t cover every feature and facility in the MVC Framework. To progress, we’ll now take a more systematic look at each aspect of the framework. In Chapter 8, you’ll learn more about the core routing system. In Chapters 9 and 10, you’ll see what’s on offer as you build controllers and actions. Chapter 11 focuses on the framework’s built-in view engine. Chapter 12 describes what ASP.NET MVC does for your models. The rest of the book considers other common web development tasks and scenarios, including security and deployment.

But hang on a minute—to make sure we don’t get lost in the small-print details of each MVC component, let’s take stock of the bigger picture. This chapter will summarize the overall landscape of MVC applications: the default project structure and naming conventions you must follow. You’ll also get a condensed view of the entire request processing pipeline, showing how all the framework components work together.

**Note**: More advanced readers who are already comfortable with the ASP.NET MVC project structure, and how IIS, routing, controllers, and views all fit together at runtime, can skip ahead to the next chapter where we’ll begin exploring the depths of the framework and its extensibility. This short chapter is intended to bridge any gaps in understanding for readers newer to ASP.NET MVC and ASP.NET generally.

Developing MVC Applications in Visual Studio

When you use Visual Studio to create a brand new ASP.NET MVC 2 project, the File ➤ New ➤ Project menu gives you two possible starting points:

- ASP.NET MVC 2 Web Application
- ASP.NET MVC 2 Empty Web Application (note that to see this option, you’ll need to open the Web category inside the File ➤ New ➤ Project dialog’s Project Types or Installed Templates list)

The first option sets up a relatively large initial set of files and folders matching those shown in Figure 7–1. This suggested project structure is supposed to help you get started; it provides a skeleton application with a simplistic way of performing navigation, user registration, and authentication. The
second option gives you a smaller number of initial files and folders; it tries to stay out of your way by giving you just the minimal structure that almost all ASP.NET MVC projects will actually want to keep. Some of these initial items have special roles hard-coded into the MVC Framework (and are subject to predetermined naming conventions), while others are merely suggestions for how to structure your project. These roles and rules are described in Table 7–1.

![Solution Explorer](image)

**Figure 7–1.** Solution Explorer immediately after creating a new ASP.NET MVC 2 Web Application project and enabling Show All Files. Note that the Empty project template does not create any of these controllers or view files by default; it mainly creates empty folders.