



第1章

初识ASP.NET

内容导读

ASP.NET 是微软发布的新一代主流企业级 Web 应用开发技术平台，是目前最热门的 Web 开发技术之一，可提供构建基于企业级服务器的 Web 应用程序所必需的所有服务。ASP.NET 是在 .NET Framework 的基础上构建的，因此所有 .NET Framework 功能都适用于 ASP.NET 应用程序。可使用与公共语言运行库 (CLR) 兼容的任何语言 (包括 Visual Basic、C++ 和 C# 等) 编写应用程序。

本章将讲解 Web 基础知识、ASP.NET 框架体系以及其运行环境配置等知识。目的是创建 Web 程序并能够对 Web 知识有一个清晰的认识。

通过本章的学习，读者应该重点掌握 IIS Web 服务器的配置和 Visual Studio 2015 集成开发环境的应用等，并能够利用其开发环境创建第一个 ASP.NET Web 的应用程序。

学习目标

- ◆ 了解什么是 Web
- ◆ 熟悉 .NET Framework 的概念
- ◆ 熟悉 Visual Studio 2015 的基本操作
- ◆ 掌握如何创建 ASP.NET 应用程序

课时安排

- ◆ 理论学习 1 课时
- ◆ 上机操作 2 课时





1.1 Web 开发技术基础

Web 是 World Wide Web 的缩写, 也称为 WWW 或万维网。目前, 它是 Internet 上应用最为广泛、最为重要的信息服务类型。

1.1.1 Web 的基本概念

Web 被人们誉为 20 世纪最伟大的发明之一。万维网使得全世界的人们以史无前例的巨大规模相互交流。无论相距如何遥远的人们, 甚至是不同年代的人们都可以通过网络来交流他们之间的信息从而实现沟通, 人们可以用历史上从来没有过的低投入实现数据共享。

最初 Web 起源于 CERN(欧洲粒子物理实验室), 由从事高性能物理研究的科学家提出设想、发明, 并最终给现代社会带来了巨大的影响。最初 CERN 的科学家是为了让散布于世界各地的高性能物理研究人员能够及时地传递信息、共享研究成果和思想, 1989 年 CERN 的 Tim Berners Lee 提出 HyperText(超文本) 的设想。1991 年, Web 的第一个技术标准问世并获得了巨大的成功。

风靡全球的 WWW 服务主要是以一个个网页呈现出来的, 所谓 Web 网页也就是大家在浏览器上见到的画面, 一般在输入网址进入站点后所看的第一个页面称为主页。主页通常用来作为一个站点的目录或索引。而 Web 网站是一组相关网页的集合, 也就是说, 设计制作了几个网页, 并且经过组织规划, 让网页彼此相连, 然后让连上 Internet 的人们都能看到, 这样的结构就称为 Web 网站。

通过 Internet, Web 将全世界联网的不同网址上的相关信息有机地编织在一起, 以超文本传输协议 (Hyper Text Transfer Protocol, HTTP) 为基础协议进行数据传输。Web 采用客户端 / 服务器工作模式, 如图 1.1 所示。

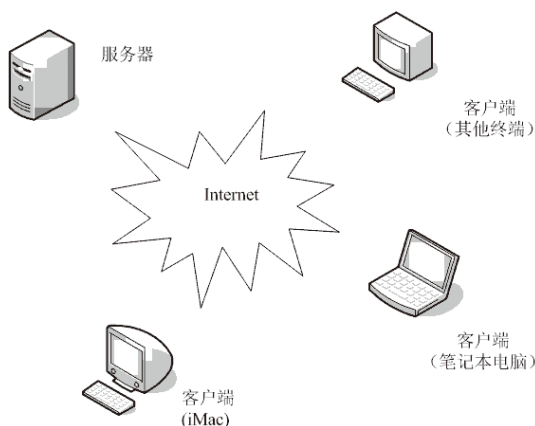


图 1.1 Web 的工作模式

服务器中的信息以页面 (或称 Web 网页) 的形式存储, 而这些页面则采用超文本标记语言 (Hyper Text Markup Language, HTML) 来对信息进行组织, 并通过超级链接将它们连

接起来。服务器的地址以及各个页面的地址信息被称为统一资源定位器 (Uniform Resource Locator, URL), 如 `http://www. Microsoft.com /windows/ default.aspx`。其中, `http://` 表示使用 HTTP 协议, `www. Microsoft.com` 表示 Microsoft 公司的 Web 服务器地址, `/windows` 表示服务器上网站根目录下面的 windows 目录 (或者是一个虚拟的目录, 但是映射了另外一个真实的目录), `/default.aspx` 表示该目录下的一个文件。所有的 Web 站点都有一个辨识自己的唯一 IP(Internet Protocol, 网际协议) 地址, 该地址由 4 个数字序列 (最大为 3 位十进制数, 值为 0~255) 组成, 中间由点号分隔, 如 127.0.0.1。其实网站的 URL 就是某个 IP 地址的映射, 使用 URL 的好处就是便于记忆。Internet 中的域名服务程序 DNS (Domain Name Services) 会自动把 URL 和 IP 地址进行转换。

客户端可以是笔记本电脑、iMac 电话、个人电话、掌上电脑以及其他终端设备。在客户端上一般都有一个 Web 浏览器软件, 如 Microsoft 公司的 Internet Explorer、Netscape 的 Navigator、掌上电脑中的 Opera 浏览器等。用户在客户端通过浏览器向服务器提出要访问的 Web 页面 URL, DNS 服务器会把要访问的 URL 指向服务器, 服务器接收到客户的请求后查找用户指定的页面, 如果没有找到该页面, 就返回一个错误信息; 否则将该页面的内容返回给客户端。客户端接收到服务器的响应后, 便将结果显示在浏览器上。

Web 页面在 Internet 上的传输是通过超文本传输协议 (HTTP) 实现的。超文本传输协议是一个应用程序协议, 允许浏览器和服务器相互通信, 来回传送数据。Web 页的所有请求和服务器发送的所有响应都是在浏览器和服务器之间传送的 HTTP 信息。

1.1.2 静态网页技术

当人们浏览 Internet 时, 会看到许多静态 Web 页面。静态网页在制作完成并发布后, 网页内容和外观对于任何浏览者无论何时或以何种方式访问都将保持不变。从本质上说, 这种类型的 Web 页面是由一些 HTML 代码组成的页面, 且这些代码可以直接通过文本编辑器或 Web 页面编辑器输入, 并保存为扩展名为 .htm 或 .html 的文件。因此, 在用户访问 Web 页面之前, Web 页面的作者已经用 HTML 完全确定了页面的具体内容。静态 Web 页面通常非常容易识别, 它的内容 (文本、图像、超链接等) 和外观总是保持不变, 它并不考虑谁在访问页面、何时访问页面、如何进入页面以及其他因素。

【例 1.1】 静态网页示例。

编写以下简单 HTML 代码, 为 Web 站点创建名为 Default.html 的页面。

```
<html>
<head><title> 欢迎进入 ASP.NET 的世界 </title></head>
<body>
<h1> 欢迎进入 ASP.NET 的世界 </h1>
<br><br>
<a HREF="contents.html"> 详细内容请点击这里 </a>.
<br><br>
```

当您在浏览的过程当中遇到问题, 请与我联系



```
<a href="mailto:happy@163.com"> 发邮件给我 </a>.  
</body>  
</html>
```

当用户进入站点访问该页面时，会得到如图 1.2 所示的结果。



图 1.2 静态页面的显示效果

在用户发出访问页面的请求之前，页面的内容已经确定。实际上，Web 管理员将代码以 .html 文件保存到磁盘时，就确定了该页面的内容。

1.1.3 动态网页技术

随着 Internet 的发展，静态页面已经不能满足人们的需要，人们需要从 Internet 上获取更多的多媒体信息，期望能够在浏览 Web 页面时看到更为吸引人的页面以及新闻信息，更希望与网站之间产生实时互动，得到更好的上网体验。特别是随着电子商务的发展，人们需要更为灵活、及时的互动 Web 技术，因此动态网页技术也就应运而生。

动态网页技术就是指网页内含有在服务器端执行的程序代码，当客户端向服务器端提出请求时，程序的代码会先在服务器端执行，然后再将 Web 服务器端执行的结果传送给浏览器。Web 服务器端执行的程序一般有 CGI、ASP、ASP.NET、JSP、PHP 等。与静态网页技术的区别如表 1.1 所示。

表 1.1 静态网页与动态网页的区别

比较项	静态网页	动态网页
内容	网页内容固定	网页内容动态生成
后缀	.htm 等	.asp、.aspx、.shtm、.php、.cgi 等
优点	无须系统实时生成、网页风格灵活多样	日常维护简单、更改结构方便、实时交互性能强

续表

比较项	静态网页	动态网页
缺点	交互性能较差、日常维护烦琐	需要大量的系统资源合成网页、 需要更强大的编程能力
数据库	不支持	支持

1.1.4 B/S 架构体系

目前在应用开发领域中,从软件开发体系上分类,可分成三大体系,即基于浏览器的 B/S(Browser/Server) 架构、基于客户端的 C/S(Client/Server) 架构、基于嵌入式系统的开发架构。基于 B/S 架构的编程体系,目前主要采用 3 种服务器端语言,即 JSP(Java Server Pages)、PHP(Personal Home Page) 和 ASP.NET。这 3 种语言构成 3 种常用应用开发组合,即 JSP+Oracle 体系、PHP+MySQL 体系以及 ASP.NET+SQL Server 体系。

B/S 架构编程语言分成浏览器端编程语言和服务端编程语言。浏览器端包括超文本标记语言 (HyperText Markup Language, HTML)、级联样式表单 (Cascading Style Sheets, CSS)、JavaScript 语言和 VBScript 语言。所谓的浏览器端编程语言就是这些语言均被浏览器解释执行。HTML 和 CSS 是由浏览器解释的,JavaScript 语言和 VBScript 语言也是在浏览器上执行的。

为了实现一些复杂的操作,如连接数据库、操作文件等,需要使用服务器端编程语言。目前主要是 3P(ASP.NET、JSP 和 PHP) 技术。ASP.NET 是 Microsoft(微软)公司推出的,在这 3 种语言中是用得最为广泛的一种。JSP 是 Sun 公司推出的,是 J2EE(Java 2 Enterprise Edition, Java2 企业版)13 种核心技术中重要的一种。PHP 在 1999 年的下半年和 2000 年用得非常广泛,因为 Linux+PHP+MySQL(一种中小型数据库管理系统)构成全免费的而且非常稳定的应用平台,这 3 种语言是目前应用开发体系的主流。

数据库支持是必需的,目前应用领域的数据库系统全部采用关系型数据库 (Relation Database Management System, RDBMS)。在企业级开发领域中,目前主要采用三大厂商的数据库关系系统,即微软公司的 SQL Server、Oracle 公司的 Oracle 以及 IBM 公司的 DB2。



1.2 .NET Framework 与 ASP.NET

ASP.NET 是建立在 .NET Framework(又叫 .NET 框架)的公共语言运行库上的编程框架,可用于在服务器上生成功能强大的 Web 应用程序。

1.2.1 .NET Framework 简介

微软公司将未来计算机软件的发展状况设想命名为 .NET, 它认为未来的计算机软件将是多个应用程序通过网络以一种分布式的方式计算的世界。在此基础上为了让不同的编程



语言都能够实现这种互通的编程方式，Microsoft 推出了 .NET。Microsoft 的 .NET 其实是一项非常庞大的计划，也是 Microsoft 发展的战略核心。Microsoft 公司前首席执行官兼总裁曾说：“.NET 代表了一个集合、一个环境、一个编程的基本结构，作为一个平台来支持下一代互联网。.NET 也是一个用户环境，是一组基本的用户服务，可以作用于客户端、服务器端或任何地方，与改编成的模式具有很好的一致性，并有所创新。因此，它不仅是一个用户体验，而且是开发人员体验的集合。”

.NET 的核心是 .NET 框架，用英文表示为 .NET Framework。2002 年，微软推出 .NET Framework 1.0 版。.NET 框架平台不同于以往的编程语言，它实质上是一个技术框架，即架设一个开发 VB.NET、C#、ASP.NET 等应用程序的总框架。在 .NET 框架中主要包括公共语言运行库、基础类库、ASP.NET 和 C# 等语言运行库三大部分。

1. 公共语言运行库

公共语言运行库 (CLR) 是 .NET 框架的最底层，也是 .NET 框架的运行环境，它的主要功能是负责运行和维护用户编写的程序代码。当客户端第一次访问网页时，所访问的 Web 应用程序会被编译成一种中间语言 (Microsoft Intermediate Language, MSIL)，然后通过即时编译器将这种中间语言生成计算机可以执行的二进制代码，当用户再次访问该页时，系统就可以从中间语言直接编译成该页面请求所允许的可执行二进制机器代码。如果源代码被开发人员进行了重写，再次访问时，就需要重新把源代码编译为中间语言。正是这样的运行模式，使得应用程序可在不同环境下执行，实现跨平台性。另外，.NET 的公共语言运行库还提供了统一安全机制和系统资源统一管理功能。

2. 基础类库 (Base Class Library)

.NET 框架为开发人员提供了一个面向对象、统一的、层次化且可扩展的类库集 (API)。这正如 C++ 开发人员使用的 Microsoft 基类 (MFC) 库，Java 开发人员使用的 Java 类库，就像 Visual Basic 用户使用的 Visual Basic API 集一样，.NET 框架统一了微软以前所有使用的类库，这样大大方便了开发人员的编程。在程序设计中，不需要再直接调用底层的系统 API，简化了程序设计，同时减少了程序设计中的问题。最主要的是，.NET 类库具有更大的跨语言兼容性、错误处理能力和调试功能。

3. ASP.NET 和 C# 等语言运行库

ASP.NET 和 C# 等语言都是 .NET 框架中的顶层应用，它们之间的关系如图 1.3 所示。

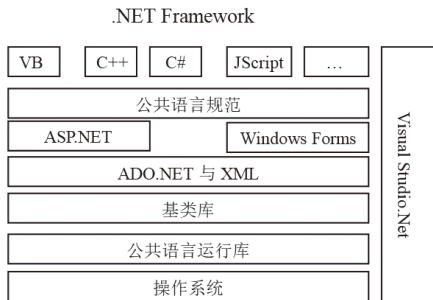


图 1.3 .NET 框架

1.2.2 ASP.NET 的基本框架

ASP.NET 是由微软公司推出的用于 Web 应用开发的全新框架，是 .NET 框架 (即 .NET Framework) 的组成部分，它从现有的 ASP(Active Server Pages) 结构体系上跨出了一大步，是对传统 ASP 技术的重大升级和更新。

ASP.NET 是一个统一的 Web 应用程序开发模型，它包括在企业级 Web 应用程序开发过程中所尽可能少的代码生成所必需的各种服务。ASP.NET 是 .NET Framework 的一部分。当开发人员编写 ASP.NET 应用程序代码时，可以方便地访问 .NET Framework 中已经封装好的类，这样可以大大降低开发难度。同样还可以使用与公共语言运行库兼容的任何语言来编写所开发应用程序的代码，这些语言包括 Microsoft Visual Basic、C#、JScript、.NET 和 J# 等。使用这些语言，可以开发利用公共语言运行库、类型安全、继承等方面优点的 ASP.NET 应用程序。ASP.NET 包括页和控件框架、ASP.NET 编译器、安全基础结构、状态管理功能、应用程序配置、运行状况监视和性能功能调试支持、XML Web Services 框架、可扩展的宿主环境和应用程序生命周期管理、可扩展的设计器环境等。

1. 页面和控件框架

ASP.NET 页面和控件框架是一种编程框架，它能够在 Web 服务器上运行，ASP.NET 页面可以被动态地生成和呈现。ASP.NET 页面可以从任何浏览器或客户端设备所请求，ASP.NET 会将所请求的页面进行编译后向浏览器请求以 HTML 标记的方式来呈现。ASP.NET 支持基于如移动电话、手持型计算机和个人数字助理 (PDA) 等 Web 设备的移动控件。

ASP.NET 页面是完全面向对象的。在 ASP.NET 网页中，HTML 元素的处理都可以使用属性、方法和事件来实现。ASP.NET 网页框架为响应在服务器上运行的代码中的客户端事件提供统一的模型，从而使开发者不必考虑基于 Web 的应用程序中固有的客户端和服务器隔离的实现细节。

ASP.NET 页面和控件框架还提供各种功能，如对网站的整体外观和感觉的控制可以通过主题和外观来实现。除了主题外，还可以定义母版页，以使应用程序中的页面布局一致。一个母版页可以定义开发人员所希望的应用程序中的所有页 (或一组页) 所具有的布局 and 标准行为。当用户请求内容页时，这些内容页与母版页合并，产生将母版页的布局与内容页中的内容组合在一起的输出。

2. ASP.NET 编译器

ASP.NET 包括一个编译器，该编译器将包括页面和控件在内的所有应用程序组件编译成一个程序集，然后 ASP.NET 宿主环境可以使用该程序集来处理用户的请求。

3. 安全基础结构

ASP.NET 提供了除 .NET 安全功能外的高级安全基础结构，以便对用户进行身份验证和授权，并执行其他与安全相关的功能。开发人员可以使用由 IIS 提供的 Windows 身份验证功能对用户进行身份验证，也可以通过自己设计的数据库与 ASP.NET Forms 身份验证和 ASP.NET 成员资格来管理身份验证。



4. 状态管理功能

ASP.NET 提供了内部状态管理功能的相关类，通过这些对象可以使开发人员能够存储客户页请求期间的信息，如客户信息或购物车的内容等，此信息可以独立于页面上的任何控件。

5. ASP.NET 配置

ASP.NET 应用程序所使用的配置系统文件，可以定义整个 Web 服务器、网站或单个应用程序的配置设置。开发人员可以随时添加或修订配置设置，且对运行的 Web 应用程序和服务具有最小的影响。ASP.NET 配置设置信息存储在基于 XML 的配置文件中。由于这些 XML 文件是 ASCII 文本文件，因此对 Web 应用程序进行配置更改比较简单。

6. 运行状况监视和性能功能

ASP.NET 具备可以监视 ASP.NET 应用程序运行状况和性能的功能。它可以报告关键事件，这些关键事件提供了有关应用程序的运行状况和错误情况信息。通过这些事件显示诊断和监视特征的组合，可以对 ASP.NET 的记录事件以及如何记录事件等方面提供高度的灵活性。

7. 调试支持

ASP.NET 利用运行库调试基础结构来提供跨语言和跨计算机的调试支持。可以调试托管和非托管对象，以及公共语言运行库和脚本语言支持的所有语言。此外，ASP.NET 页面框架提供了可以将检测消息插入 ASP.NET 网页的跟踪模式。

8. XML Web Services 框架

ASP.NET 同样提供对 XML Web Services 的支持。XML Web Services 是包含业务功能的组件，利用该业务功能，应用程序可以使用 HTTP 和 XML 消息等标准跨越防火墙来交换信息。XML Web Services 不用依靠特定的组件技术或对象调用约定。因此，用任何语言编写、使用任何组件模型并在任何操作系统上运行的程序，都可以访问 XML Web Services。

9. 可扩展的宿主环境和应用程序生命周期管理

ASP.NET 还包括一个可扩展的宿主环境，该环境控制应用程序的生命周期，即从用户首次访问此 Web 应用程序中的资源到应用程序关闭这一时间段。虽然 ASP.NET 依赖作为应用程序宿主的 IIS 服务器，但 ASP.NET 自身也提供了许多宿主功能。通过 ASP.NET 的基础结构，开发人员可以响应应用程序事件并创建自定义 HTTP 处理程序和 HTTP 处理模块。

10. 可扩展的设计器环境

ASP.NET 中提供了对创建 Web 服务器控件的设计器，如 Visual Studio 可视化开发工具的增强支持。使用工具可以为控件生成设计时用户界面，因此开发人员也可以在可视化设计工具中配置控件的属性和内容。

1.2.3 ASP.NET 的特点

ASP.NET 带来了一场网络编程的革命，开发人员可以方便、快速地开发 Web 应用程序和 Web 服务。具体地说，ASP.NET 的优点包括以下几个方面。

1. 简易性

ASP.NET 提供了很多基于常用功能的控件, 使诸如表单提交、表单验证、数据交互等常用操作变得更加简单。ASP.NET 的事务处理模型也相当简单, 类似于 VB 的 FORM 处理模型。ASP.NET 执行窗体提交和客户端身份验证到部署和站点配置变得很容易。同时, 发布、配置程序也由于 ASP.NET 的新的处理模式而更加简单。

2. 安全性

应用 Windows 系统内置的身份验证机制和基于每个应用程序的配置, 完全可以保证应用程序的安全性。ASP.NET 还为 Web 应用程序提供了各种授权和身份验证方案。开发人员可以根据应用程序的需要方便地移除、添加或替换这些方案。

3. 自定义性和可扩展性

用户可以使用自己编写的自定义组件或扩展来替换 ASP.NET 运行库中的任何子组件。ASP.NET 是基于通用语言的, 开发者可以使用任何 .NET 支持的语言 (如 C#、VB.NET、JScript 等) 来开发 ASP.NET 的程序。ASP.NET 编程语言可以选择最适合应用程序的语言, 或跨多种语言分割应用程序。

4. 强大的工具支持

ASP.NET 框架应用了 Visual Studio.NET 集成开发环境中的工具箱和设计器。Visual Studio.NET 提供了强大、高效的 .NET 程序的集成开发环境, 支持诸如所见即所得、控件拖放、编译调试等功能, 使开发 ASP.NET 程序更加快速、方便。

5. 多种性能优良的功能

ASP.NET 与其前身 ASP 最大的区别在于其不再是解释性的脚本, 而是运行于服务器端经过编译的代码, 这使早期绑定、本地优化、缓存服务等技术成为可能, 大大提高了 ASP.NET 程序的执行效率。ASP.NET 中还包括多种功能和工具, 可用来设计和实现高性能的 Web 应用程序。这些功能包括基于 ASP 经过改进的进程模型、自动编译请求的页面并在服务器上存储这些页面、ASP.NET 特定的性能计数器、Web 应用程序测试工具等。

6. 可管理性

ASP.NET 程序的所有配置都存储在基于 XML 的文件中, 这将大大简化对服务器环境和网络程序的配置过程。在部署 ASP.NET 框架应用程序时只需将必要的文件复制到服务器, 即可将 ASP.NET 框架应用程序部署到服务器, 而不需要重新启动服务器。



1.3 配置 ASP.NET 运行环境

作为 ASP.NET 的 Windows 运行环境, IIS(Internet Information Service, 互联网信息服务)服务器的配置是必不可少的。Web 项目在开发完毕后, 必须要通过 IIS 服务器的发布才可以供其他人访问。

1.3.1 配置 IIS 服务器

作为 Windows 下网页技术的发布者和服务提供者，IIS 服务器是承载整个 Web 网站对外提供服务的重要程序，所有的 Web 页面也都是在这个软件的管理下为用户服务的。根据操作系统版本的不同，IIS 服务器的安装状况略有不同。下面就对其配置过程进行简单说明。

选择“控制面板”|“所有控制面板项”|“管理工具”，查看各自的 IIS 服务器是否已经安装成功，如图 1.4 所示。

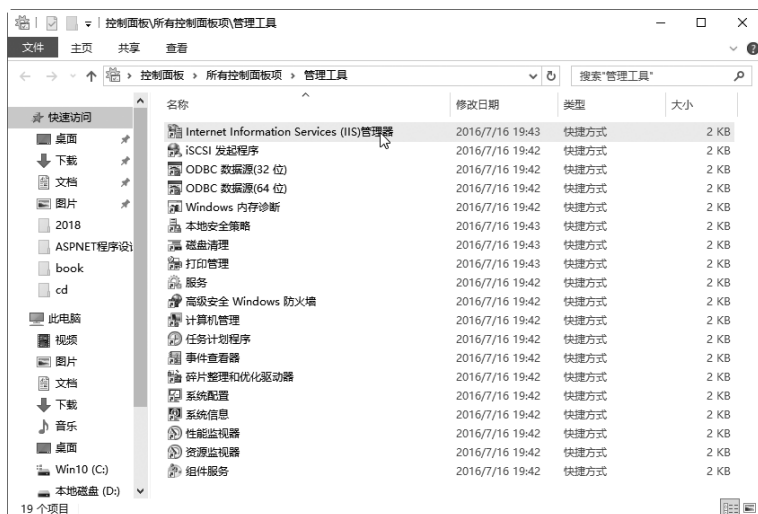


图 1.4 查看 IIS 是否安装

若没有安装，则需要安装 IIS 服务器组件。具体步骤如下：进入“控制面板”，选择“程序和功能”，如图 1.5 所示。

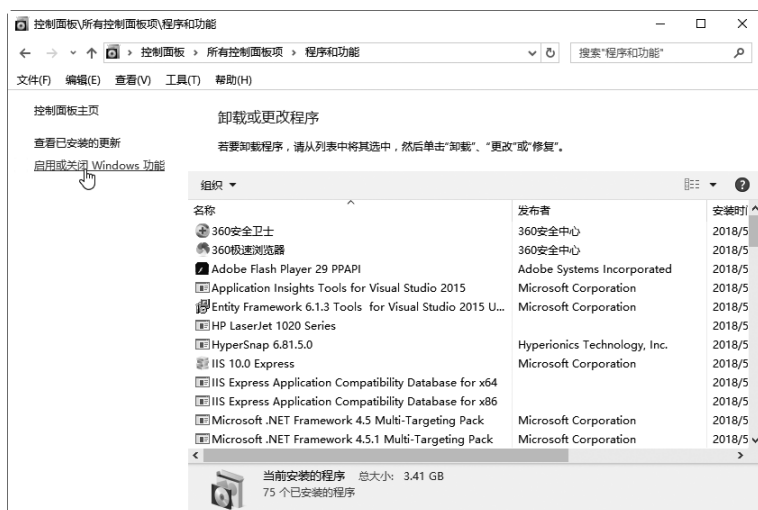


图 1.5 启用或关闭 Windows 功能

单击窗口左边的“启用或关闭 Windows 功能”链接，在弹出的窗口中找到并选择 Internet Information Services 选项，如图 1.6 所示。

将 Windows 10 安装盘放入光驱，单击“确定”按钮就可以实现 IIS 的安装。图 1.7 显示安装成功。

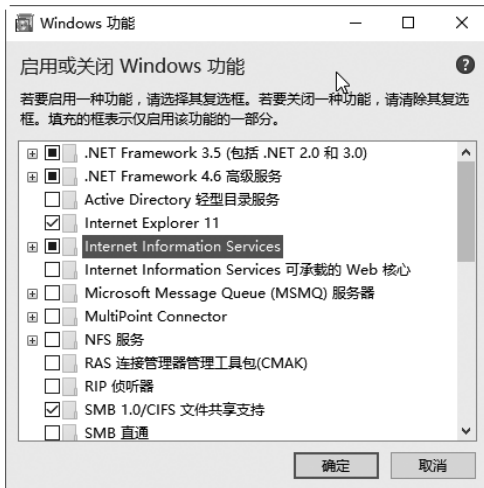


图 1.6 安装 IIS

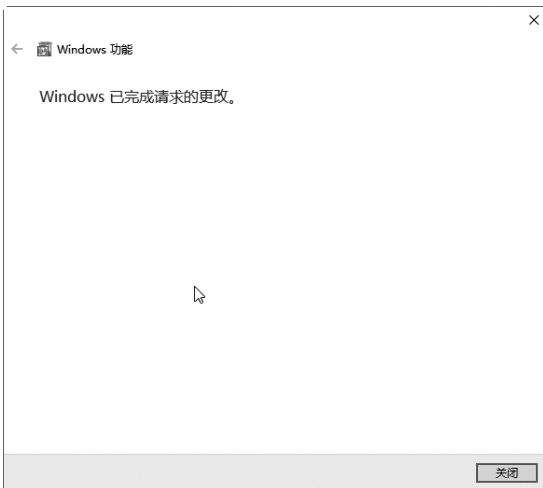


图 1.7 IIS 安装成功

在安装完毕后，就可以对 IIS Web 服务器进行配置了。选择“控制面板”|“管理工具”，双击 IIS，进入 IIS 管理器，如图 1.8 所示。



图 1.8 IIS 管理器

打开后单击窗口左边网站节点，展开“网站”，选择 Default Web Site 选项，然后选择窗口右边的“基本设置”，可以对 IIS 进行基本设置管理，如配置和添加 Web 项目所在的路径等，如图 1.9 所示。

配置网站主页。打开网站站点的 IIS 管理界面，双击“默认文档”选项，从打开的窗口中就可以配置添加该网站所要运行的初始网站主页，如图 1.10 所示。



图 1.9 Web 项目目录配置

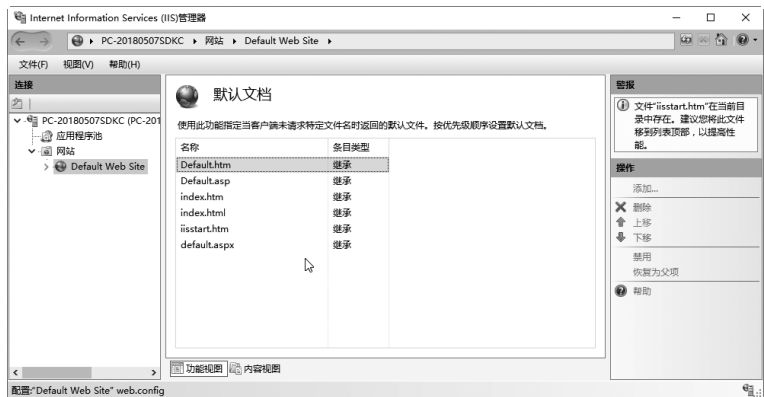


图 1.10 配置网站初始页面

配置身份验证。打开网站站点的 IIS 管理界面，双击“身份验证”选项，就可以进行配置身份验证。这里身份验证有 Forms 身份验证、ASP.NET 模拟、匿名身份验证 3 种验证方式，如图 1.11 所示。



图 1.11 身份验证配置

1.3.2 Visual Studio 开发环境介绍

下面介绍 Visual Studio 集成开发环境 (IDE)。Visual Studio 是一套基于组件的软件开发工具和其他技术，可用于构建功能强大、性能出众的应用程序。Visual Studio 不但提供了高效的开发环境，还提供了功能强大的帮助工具。Visual Studio 2015 内部代号为 Visual Studio “14”，发布时间是 2014 年 11 月 13 日。Visual Studio 2015 平台可帮助开发人员打造跨平台的应用程序，从 Windows 到 Linux 甚至 iOS 和 Android。

在安装 Visual Studio 2015 之前要确定所使用计算机的软硬件配置情况，看看能否达到基本配置的要求，以便正确安装并全面使用 Visual Studio 2015 强大的功能。

Visual Studio Professional 2015 (含 Update 3) 的硬件要求如下。

- ◎ 1.6 GHz 或更快的处理器。
- ◎ 1 GB 的 RAM(如果在虚拟机上运行则需 1.5 GB)。
- ◎ 10 GB 可用硬盘空间。
- ◎ 转速为 5400 r/min 的硬盘驱动器。
- ◎ 支持 DirectX 9 的视频卡 (1024×768 或更高分辨率)。

Visual Studio Professional 2015 (含 Update 3) 支持以下操作系统。

- ◎ Windows 10。
- ◎ Windows 8.1。
- ◎ Windows 8。
- ◎ Windows 7 SP1。
- ◎ Windows Server 2012 R2。
- ◎ Windows Server 2012。
- ◎ Windows Server 2008 R2 SP1。

Visual Studio 2015 包括以下新功能。

- ◎ 新的、灵活的跨平台运行时。
- ◎ 新的模块式 HTTP 请求管道。
- ◎ 云就绪环境配置。
- ◎ 将 MVC、Web API 和网页结合在一起的编程模型。
- ◎ 能够在不重新生成项目的情况下查看更改。
- ◎ 能并行运行多个 .NET Framework 版本。
- ◎ 能够自承载或在 IIS 上承载。
- ◎ Visual Studio 2015 中的新工具，包括 Grunt、Gulp、Bower 和 NPM 集成。



© GitHub 中的开放源代码。

2002 年，微软取消了 Visual FoxPro、Visual InterDev、Visual J++ 等，推出了 Visual Studio 和 .NET Framework 1.0 以及一门新的语言 C#。基于 .NET Framework 的 Visual Studio 融合了 3 种开发语言，即 C#、VB.NET 和 VC++.NET，使得各种语言的开发环境使用几乎相同的 IDE 集成环境，并且开发的程序都具有和平台无关的特性。随后，微软在此基础上又推出 Visual Studio 2003、Visual Studio 2005、Visual Studio 2008 等版本。

作为微软跨平台新战略下的开发工具，Visual Studio 2015 支持开发人员编写跨平台应用程序，从 Windows 到 Mac、Linux、甚至是编写 iOS 和 Android 代码。此外，微软还为此发布了比谷歌自家性能更优异的 Android 模拟器。微软进一步扩大了 .NET 框架的可应用范围，开发人员可以利用通用的代码库在 Windows/Mac/Linux 三大操作系统上快速开发自己的应用软件，这无疑是非常好的消息。如果微软能够将 .NET 做到完美跨三大平台，其发展势头将不可估量。

除了桌面操作系统外，移动开发也是微软最看重的一点。Visual Studio 2015 不仅支持 Windows Phone 的开发，还支持 iOS 以及 Android 移动端系统的应用开发。为此，Visual Studio 2015 直接内置了 Android 模拟器——Visual Studio Emulator for Android。而 iOS 开发目前没有提供模拟器，还需要使用 iPhone 真机进行测试。但无论如何，这都是一个非常大的进步。

Visual Studio 2015 包含了名为“Roslyn”的 .NET 编译器平台 (compiler platform)，现在不仅可以开发 Windows 应用，还可支持 C#、C++、Python、Visual Basic、Node.js、HTML、JavaScript 等编程语言的开发。

微软为了让 Windows 拥有更加繁荣的生态环境，决定将之前售价高昂的 Visual Studio 开发工具最大限度地免费化。从 Visual Studio 2013 开始便推出了一个全新的版本——社区版。Visual Studio 2015 社区版比以前免费的 Express 版本更高级，但依然免费提供给单个开发人员以及科研、教育和小型专业团队，并且开放源代码项目。

Visual Studio IDE 中编程人员可通过菜单栏访问大多数用来控制开发环境的命令。菜单和命令的工作方式与所有基于 Windows 的程序相同，可以使用鼠标和键盘来访问它们。菜单栏的下方是标准工具栏，这个工具栏中的按钮用于快速执行命令和控制 Visual Studio IDE。在 IDE 中右击工具栏可显示其完整列表，窗口的底部是 Windows 任务栏，如图 1.12 所示。

在 Visual Studio IDE 中，主要工具包括设计器、解决方案资源管理器、属性窗口和工具箱。此外，还会出现“服务器资源管理器”和“对象浏览器”等特殊工具，它们在 IDE 中也会以选项卡形式出现。因为每个开发人员的偏好设置不同，所以界面设置也有所不同。

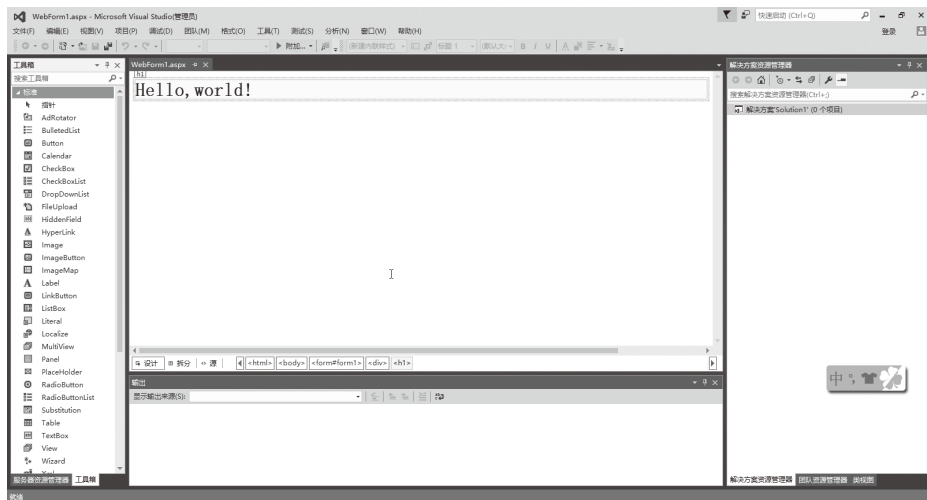


图 1.12 Visual Studio 2015 的开发环境



1.4 第一个 ASP.NET Web 程序

使用 Visual Studio 2015 开发环境可以非常轻松地创建 ASP.NET Web 应用程序。下面通过本书的第一个 ASP.NET Web 应用程序来介绍如何使用 Visual Studio 2015 创建 Web 应用程序。

1.4.1 创建第一个 ASP.NET Web 程序

【例 1.2】创建第一个 ASP.NET Web 程序。

创建 ASP.NET 程序的操作过程大致如下。

(1) 在 Visual Studio 2015 开发环境中选择菜单“文件”|“新建”|“项目”命令，如图 1.13 所示。

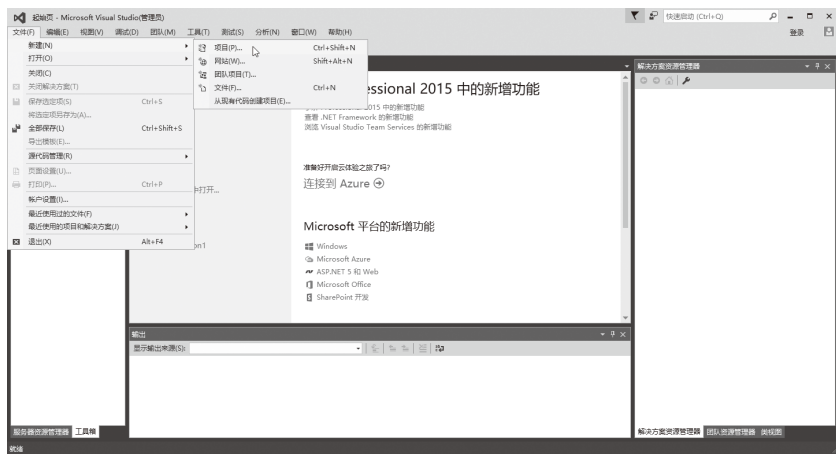


图 1.13 新建项目

(2) 打开“新建项目”对话框，在对话框左边选择“Visual C#”下面的 Web 选项，在窗口模板中选择“ASP.NET Web 应用程序”，再对其位置、项目名称及 .NET Framework(这里选择 4.5 版本)等选项进行设置，如图 1.14 所示。

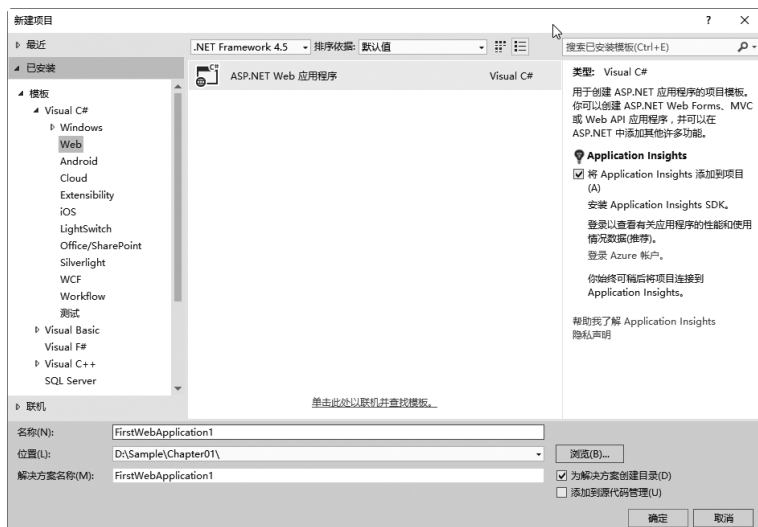


图 1.14 选择新建项目类型

(3) 设置完成后，单击“确定”按钮，进入 ASP.NET “选择模板”界面，如图 1.15 所示。

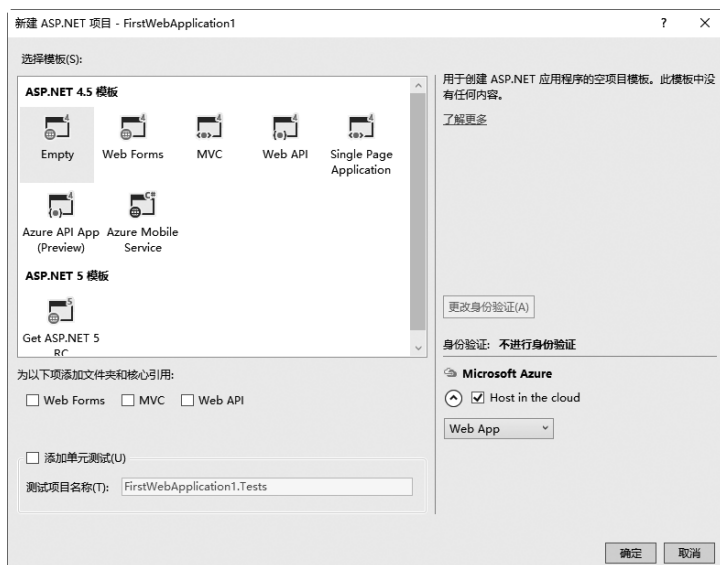


图 1.15 ASP.NET 模板选择

(4) 选择默认的 Empty 选项，单击“确定”按钮，系统将根据编程人员选择模板创建解决方案及其指定的项目目录和文件，如图 1.16 所示。

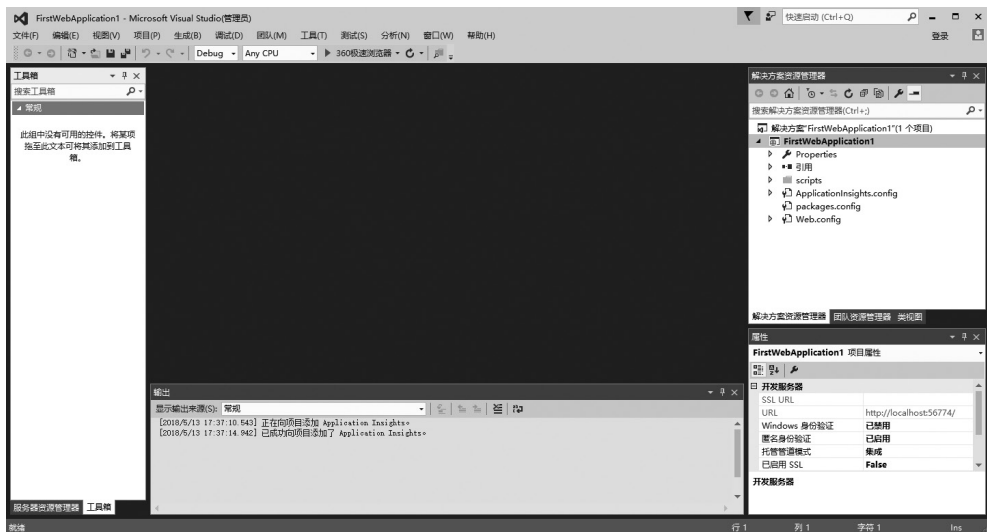


图 1.16 项目生成

在项目中添加网页文件。在菜单栏中选择“项目”|“添加新项”命令，弹出“添加新项”对话框；选择“Web 窗体”选项，在下面窗体“名称”中输入网页文件名称，系统默认为“WebForm1.aspx”，如图 1.17 所示。



图 1.17 在项目中添加网页文件

单击“添加”按钮，进入程序设计界面，页面设计视图有 3 种，即设计视图、拆分视图、源视图，这里默认打开源视图，如图 1.18 所示。

单击“设计”按钮，进入设计视图。选择菜单“视图”|“工具箱”命令，弹出“工具箱”。从工具箱中分别拖曳一个 Label 控件和一个 Button 控件到页面中，如图 1.19 所示。

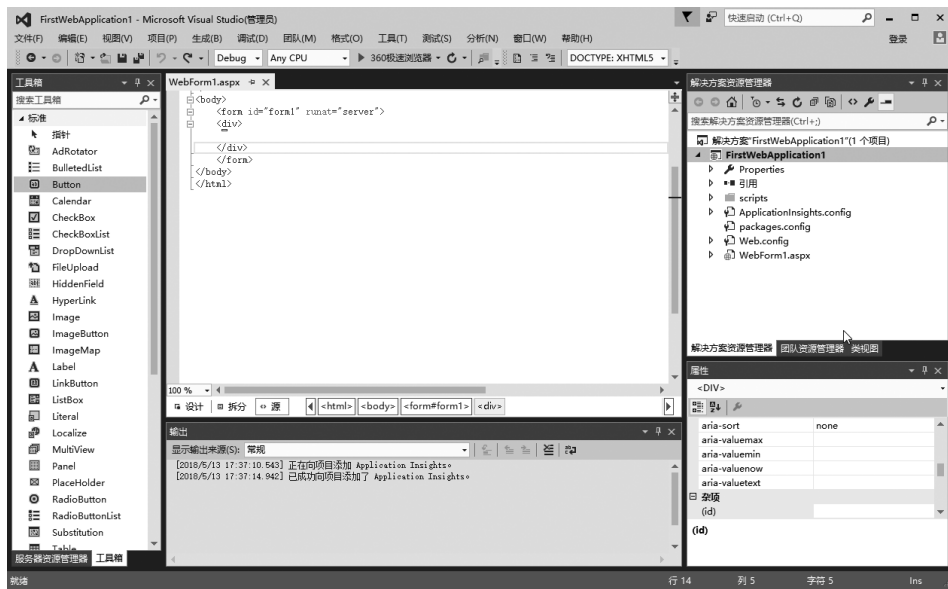


图 1.18 源代码视图

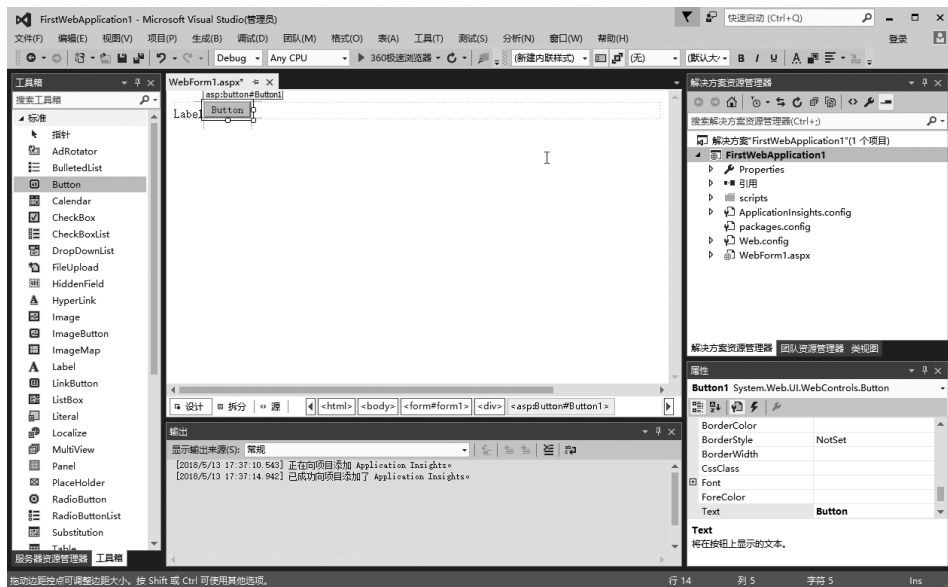


图 1.19 添加控件

设置控件属性。右击 Button 控件，在弹出的快捷菜单中选择“属性”命令，打开“属性”对话框；选择 Text 属性，将其值修改为“单击”，如图 1.20 所示。

添加事件，编写程序。双击 Button 控件，进入编写代码页面，编写 Button1 按钮的 Click 事件代码，如图 1.21 所示。

```
protected void Button1_Click(object sender, EventArgs e)
{
    Label1.Text = " 第一个 ASP.NET 应用程序！ ";
}
```

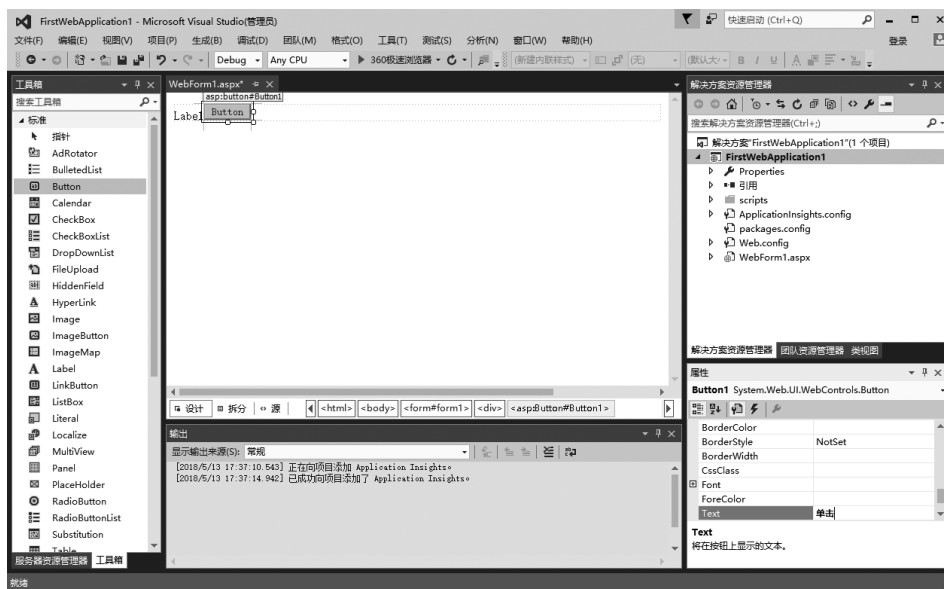


图 1.20 修改控件属性

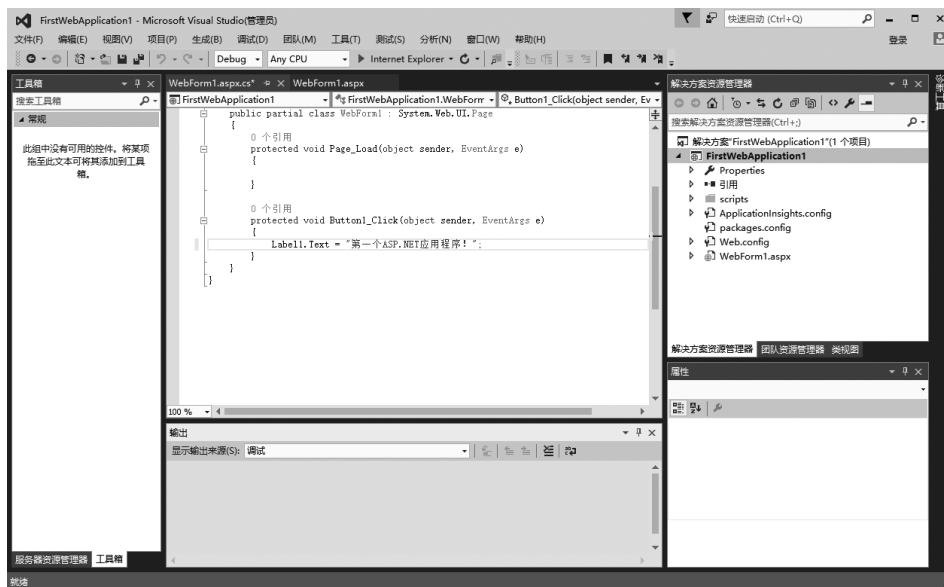


图 1.21 编写事件代码

保存项目。在菜单栏中选择“文件”|“全部保存”命令，这样就可以将所有设计的网页和编辑过的代码全部保存起来。

1.4.2 编译、运行并调试应用程序

编译、运行并调试程序是开发人员检测程序是否有错误的非常重要的功能，这里将通过实例来介绍如何调试 Web 应用程序。

(1) 在创建好 ASP.NET 网站项目后，选择菜单栏中的“生成”|“生成解决方案”命令，在“输出”窗口会显示生成解决方案的结果，如图 1.22 所示。

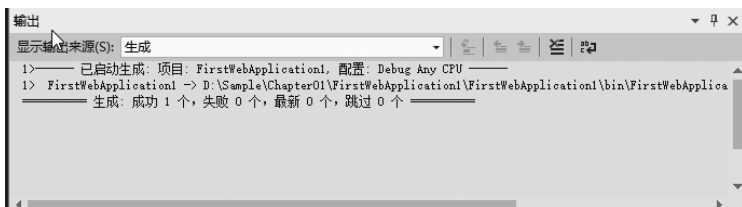


图 1.22 生成解决方案

(2) 调试、运行。选择菜单栏中的“调试”|“开始调试”命令，或直接按 F5 快捷键，即可在浏览器运行该项目；单击浏览器中的“单击”按钮，会显示“第一个 ASP.NET 应用程序！”文本，如图 1.23 所示。

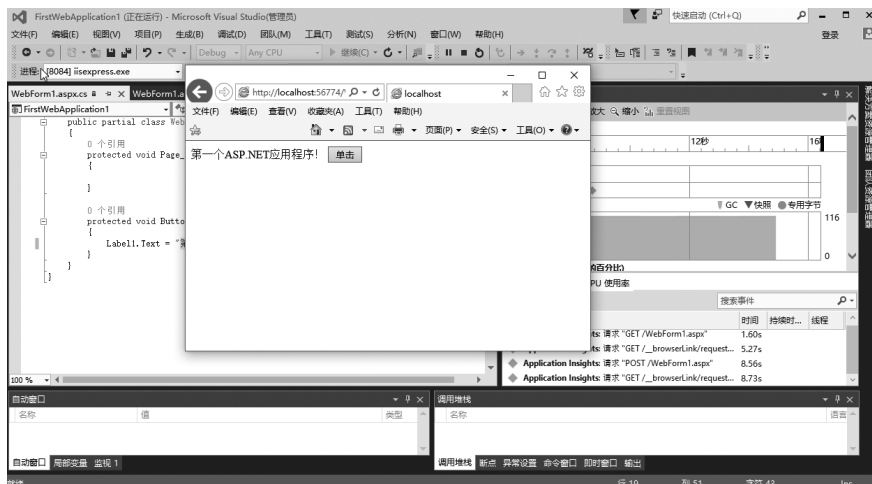


图 1.23 运行效果

(3) 为了程序员开发的方便，当程序出现问题时，可以调试该程序。在 Web 窗体上右击，在弹出的快捷菜单中选择“查看代码”命令，如图 1.24 所示。

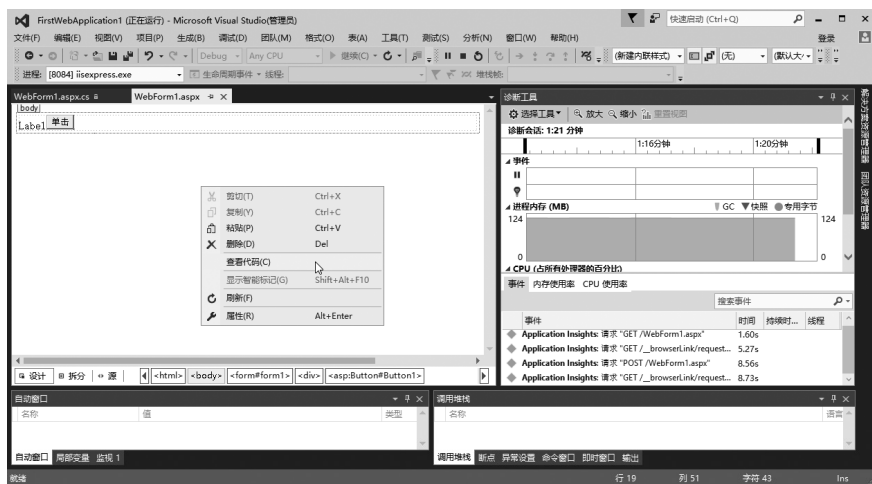


图 1.24 查看代码

(4) 在代码页面中设定一个断点，如图 1.25 所示。

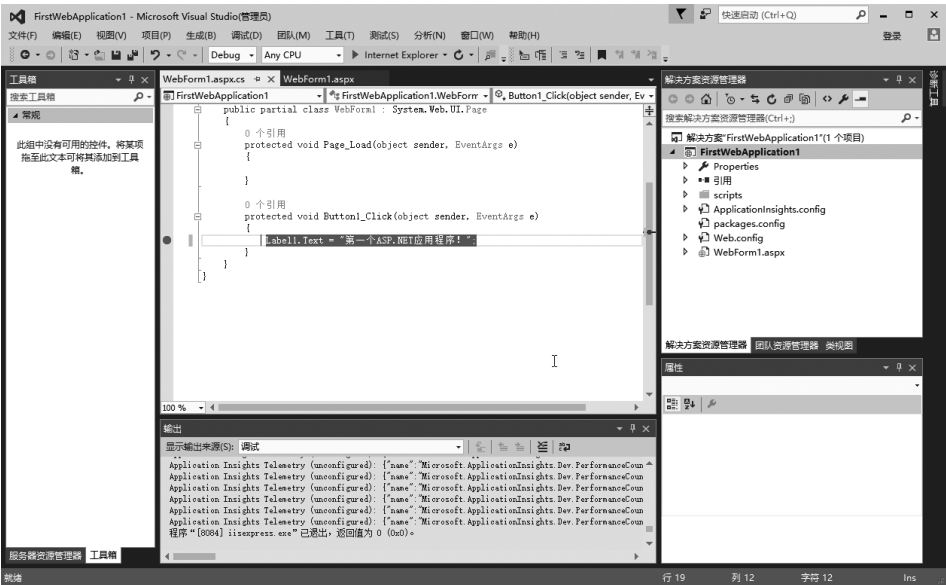


图 1.25 设定断点

(5) 选择菜单栏中的“调试”|“逐语句”或者“逐过程”命令，来调试该 Web 应用程序，如图 1.26 所示。

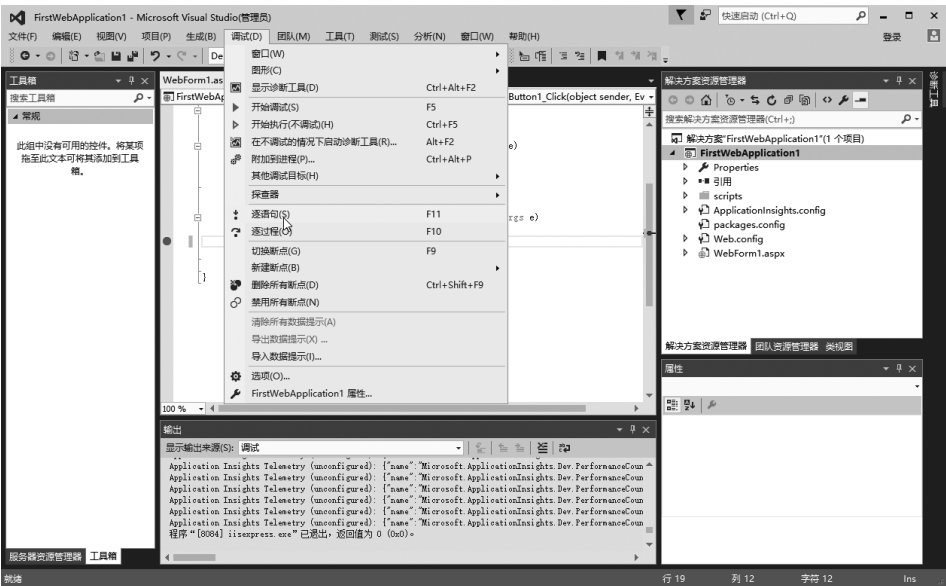


图 1.26 逐语句调试程序

这样就可以随着断点的前进来查看任何一个已经设定好的变量值，从而发现程序出现的问题，实现程序的调试过程。



强化练习

本章主要介绍了 Web 基础知识、ASP.NET 框架体系以及其运行环境配置等知识。在此基础上,读者能够创建 Web 程序并对 Web 知识有一个清晰的认识。通过对本章内容的学习,读者应该重点掌握 IIS Web 服务器的配置和 Visual Studio 2015 集成开发环境的应用等,并能够利用其开发环境创建第一个 ASP.NET Web 应用程序。

练习 1:

练习配置 IIS 服务器。在进行该练习时,先查看本机是否已经安装了 Web 服务器,如果没有安装先自行安装,然后打开 IIS 进行配置。该练习参考 1.3.1 节讲解的知识。

练习 2:

在计算机中安装 Visual Studio 2015 开发环境并熟悉开发主界面、菜单栏的命令。

练习 3:

练习利用 Visual Studio 2015 开发环境创建 ASP.NET 应用程序。

在 Visual Studio 2015 开发环境中新建一个 Web 项目并编译、运行。该练习参考 1.4 节讲解的知识。



常见疑难解答

问: 解释什么是 Web ?

答: Web 是 World Wide Web 的缩写,也称为 WWW 或万维网。目前,它是 Internet 上应用最为广泛、最为重要的信息服务类型。

问: 静态网页技术和动态网页技术有什么区别?

答: 静态网页在制作完成并发布后,网页内容和外观对于任何浏览者无论何时或以何种方式访问都将保持不变。动态网页内容会根据访问者、访问时间或方式的变化而动态生成。

问: .NET Framework 是什么?

答: .NET Framework 是 Microsoft 公司 .NET 技术的核心。.NET Framework 平台不同于以往的编程语言,它实质上是一个技术框架,即架设一个开发 VB.NET、C#、ASP.NET 等应用程序的总框架。在 .NET 框架中主要包括公共语言运行库、基础类库和 VB.NET、C#、ASP.NET 语言运行库三大部分。

问: 什么是 IIS 服务器? IIS 服务器是用来干什么的?

答: IIS 服务器是 Internet Information Service(互联网信息服务) 的简称。作为 Windows 下网页技术的发布者, IIS 服务器是承载整个 Web 网站对外提供服务的重要程序,所有的 Web 页面也都是在这个软件的管理下为用户服务的。

问: Visual Studio 2015 是否支持跨平台编程?

答: 作为 Microsoft 跨平台新战略下的开发工具, Visual Studio 2015 支持开发人员编写跨平台应用程序,从 Windows 到 Mac、Linux 甚至是编写 iOS 和 Android 代码。

问: ASP.NET 的优越性有哪些?

答: ASP.NET 有简易性、安全性、自定义性和可扩展性、强大工具支持、多种性能优良的功能、可管理性等优点。