

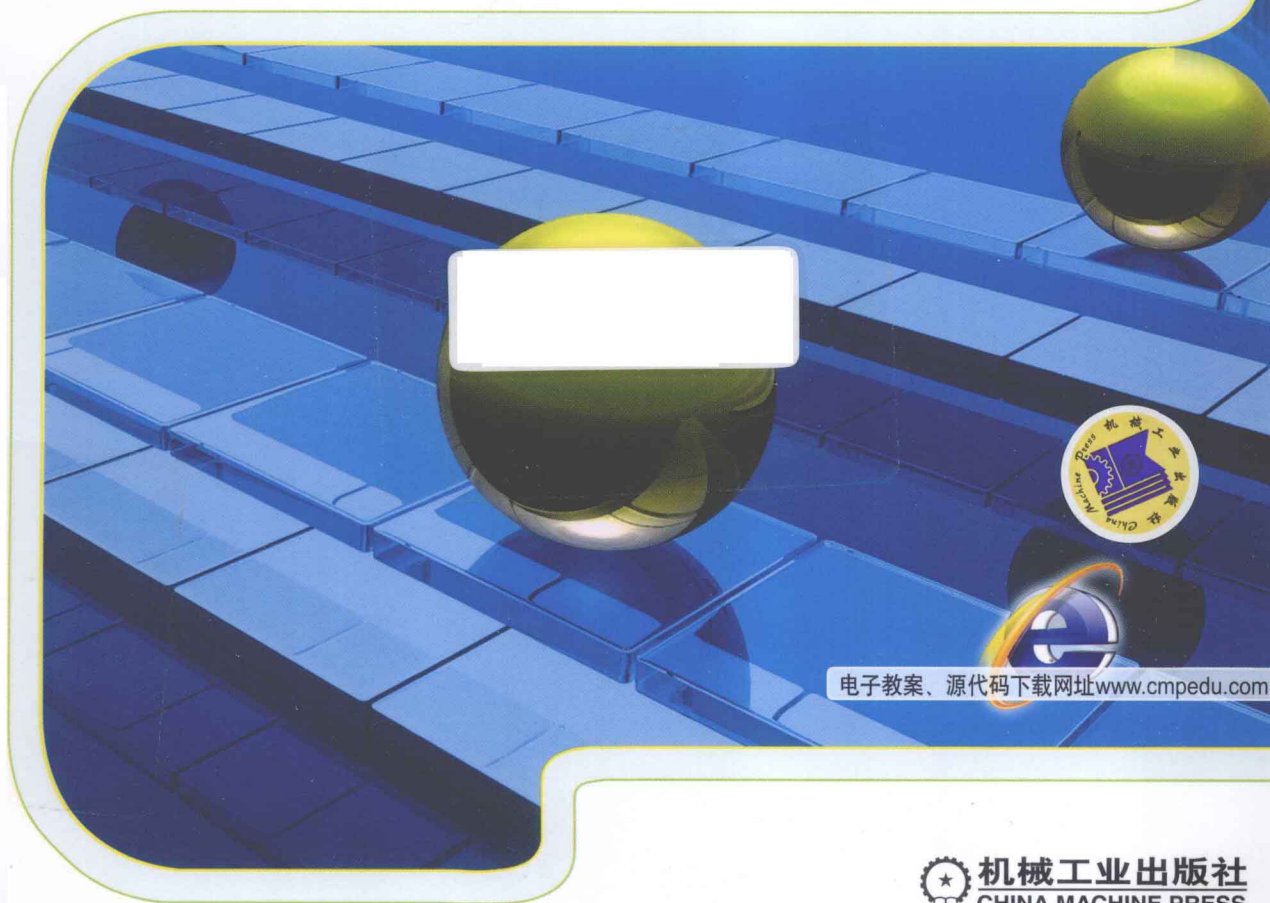


全国高等职业教育规划教材

ASP.NET应用程序开发技术

主编 章忠宪

参编 唐毅鋈 郭里城 陈子达



电子教案、源代码下载网址 www.cmpedu.com



机械工业出版社
CHINA MACHINE PRESS

全国高等职业教育规划教材

ASP.NET 应用程序开发技术

主 编 章忠宪

参 编 唐毅鏊 郭里城 陈子达



机械工业出版社

本书是作者在多年教学实践基础上,结合国家示范性建设成果编写而成的。本书使用 C#作为程序设计语言,按照典型单体技术应用划分教学单元,每个学习单元精心设计若干涵盖本单元典型技术的案例,以学习任务为中心,按照任务场景描述、任务引导、相关知识与技术说明、任务的设计与实现说明、知识与技术拓展和课业任务及指导等学习环节组织教学内容。书中还提供了一个内容翔实的综合训练项目案例,以培养学生 ASP.NET 应用项目开发能力。

书中教学案例设计典型,技术点选择实用,内容讲解详细,可作为高等职业院校计算机类各专业.NET Web 程序设计教材,也可以作为广大程序设计开发者、爱好者的自学参考书。

本书配有授课电子课件和源代码,需要的教师可登录 www.cmpedu.com 免费注册、审核通过后下载,或联系编辑索取(QQ: 1239258369, 电话: 010-88379739)。

图书在版编目(CIP)数据

ASP.NET 应用程序开发技术/章忠宪主编. —北京:机械工业出版社, 2013. 8

ISBN 978-7-111-43260-9

I. ①A… II. ①章… III. ①网页制作工具—程序设计—高等职业教育—教材 IV. ①TP393.092

中国版本图书馆 CIP 数据核字(2013)第 156262 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:鹿 征

责任印制:乔 宇

北京机工印刷厂印刷(三河市南杨庄国丰装订厂装订)

2014 年 1 月第 1 版·第 1 次印刷

184mm×260mm·15.25 印张·376 千字

0 001—3 000 册

标准书号:ISBN 978-7-111-43260-9

定价:32.80 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010)88361066 教材网:<http://www.cmpedu.com>

销售一部:(010)68326294 机工官网:<http://www.cmpbook.com>

销售二部:(010)88379649 机工官博:<http://weibo.com/cmp1952>

读者购书热线:(010)88379203 封面无防伪标均为盗版

前 言

ASP.NET 易入门、易开发、所见即所得、开发周期短、运行效率高，它构建在 .NET Framework 之上，聚合了紧密相关的多种新技术，彻底改变了从数据库访问到分布式应用程序的一切，可以开发广泛适用的、高效的 Web 应用程序。ASP.NET 已经成为当前最为完整的 Web 应用程序开发技术平台之一，成为程序员广泛的选择。

ASP.NET 支持多种开发语言，并且在 .NET 上所有语言程序都编译为 MSIL 中间语言。从这个角度讲，开发 ASP.NET Web 应用程序，选择什么语言并不重要。但是，由于 C# 从 C/C++ 进化而来，既结合了 C/C++ 的功能强大特性，又具备了 VB 的易用性，同时语法又与 Java 相近，因此本书选择 C# 作为程序设计语言。

本书侧重于解析 ASP.NET 应用开发技术，关于 C# 语言程序设计基础知识与基础技能的细节没有涉及，使用本教材，读者需要具备 C# 程序设计的基本技能。任课教师可根据教学对象的具体情况对相关基础知识技能作必要的补充。

本书按照“单体技术+综合项目”的模式组织教学单元，首先以技术技能掌握为目标划分 8 个单体技术技能学习单元，然后以技术的综合运用为目标设计了课程综合项目案例。具体到每个单体学习单元，由简单到复杂设计若干个典型的技术应用任务，以任务为驱动、以案例为载体，按照任务场景描述、任务引导、相关知识与技术说明、任务的设计与实现说明、知识与技术拓展和课外课业及指导等环节组织教学内容。而课程综合项目案例部分按照项目背景及需求分析、系统架构设计、数据库设计、系统的文件及其组织结构和功能的设计与实现等，以类似于系统设计说明书的文体组织教学内容。

由于篇幅的限制，相关的大量程序代码没有在书中列出，因此建议读者在阅读本书有关任务设计与实现的内容时，应与本书涉及的教学案例结合起来，从而达到事半功倍的效果。

本书纳入“福建省高等职业教育教材建设计划”，在编写过程中得到了福建省教育厅的大力支持，在此表示衷心的感谢！

本书由章忠宪主编，学习单元 1、3、4、9 由章忠宪编写，学习单元 5 和 7 由郭里城编写，学习单元 2 和 8 由唐毅鎏老师编写，学习单元 6 由陈子达编写。本书在教学案例设计方面还得到了福建新大陆电脑有限公司、中教信息技术有限公司和中软海晟等公司的帮助，在此表示衷心感谢！

由于水平有限，加之时间仓促，书中难免有疏漏和不妥之处，敬请专家、同仁和广大读者不吝指正。

编 者

目 录

出版说明

前言

学习单元 1 认识 ASP.NET 并设计一个 简单的 ASP.NET Web 应用程序	1
1.1 认识 .NET Framework	1
1.1.1 认识 .NET Framework 的组成	1
1.1.2 选择合适的 .NET 语言	2
1.2 认识、创建并运行 ASP.NET Web 应用程序	2
1.2.1 ASP.NET Web 应用程序运行原理	2
1.2.2 ASP.NET Web 应用程序运行环境	4
1.2.3 ASP.NET Web 应用程序构成	4
1.3 认识 ASP.NET Web 页面	5
1.3.1 ASP.NET Web 页面组成	5
1.3.2 使用 HTML 标记和 Web 控件设计页面	7
1.4 在 Web 应用程序中添加程序代码——运行逻辑	9
1.4.1 事件驱动程序设计原理	9
1.4.2 响应页面及控件事件	9
1.4.3 代码隐藏技术	11
1.5 使用 Visual Studio.NET 开发 ASP.NET 应用程序	13
1.5.1 使用 Visual Studio.NET 开发 ASP.NET 应用程序的方法	13
1.5.2 使用 Visual Studio.NET 开发一个简单的 ASP.NET 应用程序	17
1.6 知识与技能拓展——理解并使用页面的生命周期	19
1.6.1 控件的缓存事件与回发事件	19
1.6.2 页面生命周期	19
1.6.3 使用页面回发检测优化页面初始化	20
学习单元 2 设计 ASP.NET 页面	22
2.1 任务 1: 使用 ASP.NET Web 控件设计 ASP.NET Web 页面	22
2.1.1 任务场景	22
2.1.2 学习引导	22
2.1.3 相关知识与技术	22
2.1.4 任务的设计与实现	29
2.2 任务 2: 运用母版页技术设计 ASP.NET Web 页面	33
2.2.1 任务场景	33
2.2.2 学习引导	33

2.2.3	相关知识与技术	33
2.2.4	任务的设计与实现	35
2.3	任务 3: 使用页面导航技术连接 ASP.NET Web 应用程序中的页面	37
2.3.1	任务场景	37
2.3.2	学习引导	37
2.3.3	相关知识与技术	37
2.3.4	任务的设计实现	42
2.4	技术拓展: 使用第三方控件	44
2.4.1	引用第三方控件	44
2.4.2	使用 JS 日期控件	45
2.4.3	验证码控件	45
2.4.4	动态生成树形导航	45
2.5	课业任务及指导	48
学习单元 3 使用数据源控件操作数据库		50
3.1	任务 1: 使用 SqlDataSource 数据源控件操作数据库	50
3.1.1	任务场景	50
3.1.2	学习引导	51
3.1.3	相关知识与技术	51
3.1.4	任务的设计与实现	62
3.2	任务 2: 使用 GridView 和 DetailsView 的模板字段操作数据库	66
3.2.1	任务场景	66
3.2.2	学习引导	67
3.2.3	相关知识与技术	68
3.2.4	任务的设计实现	69
3.3	技术拓展: 使用存储过程优化访问数据库性能	75
3.4	课业任务及指导	76
学习单元 4 使用 ADO.NET 操作数据库		79
4.1	任务 1: 使用 DataReader 及其相关对象操作数据库	79
4.1.1	任务场景	79
4.1.2	学习引导	81
4.1.3	相关知识与技术	84
4.1.4	任务的设计与实现	97
4.2	任务 2: 运用 ObjectDataSource 数据源控件实现三层结构数据库应用程序	102
4.2.1	任务场景	102
4.2.2	学习引导	103
4.2.3	相关知识与技术	104
4.2.4	任务的设计与实现	106
4.3	技术拓展: 使用 FreeTextBox 控件编辑并获取 HTML	114
4.3.1	FreeTextBox 控件重要属性	114

4.3.2	FreeTextBox 控件软件包组成	114
4.3.3	使用 FreeTextBox 控件	115
4.4	课业任务及指导	115
学习单元 5	使用 LINQ 技术操作数据库	117
5.1	任务 1: 使用 LINQ to SQL 操作数据库	117
5.1.1	任务场景	117
5.1.2	学习引导	119
5.1.3	相关知识与技术	119
5.1.4	任务的设计与实现	126
5.2	任务 2: 使用 LinqDataSource 控件操作数据库	132
5.2.1	任务场景	132
5.2.2	学习引导	132
5.2.3	相关知识与技术	133
5.2.4	任务的设计与实现	135
5.3	课业任务及指导	137
学习单元 6	跟踪和管理应用程序及用户状态	142
6.1	任务 1: 使用 Application 对象跟踪应用程序状态	142
6.1.1	任务场景	142
6.1.2	学习引导	143
6.1.3	相关知识与技术	144
6.1.4	任务的设计与实现	147
6.2	任务 2: 使用 Session 对象或者 Cookies 对象跟踪用户	148
6.2.1	任务场景	148
6.2.2	学习引导	150
6.2.3	相关知识与技术	151
6.2.4	任务的设计与实现	155
6.3	技术拓展: 获取客户端或服务器端环境状态	157
6.3.1	获取客户端环境状态	157
6.3.2	获取服务器端环境状态	157
6.4	课业任务及指导	158
学习单元 7	保护 ASP.NET Web 应用程序	160
7.1	任务 1: 利用表单验证技术实施应用程序功能的目录级保护	160
7.1.1	任务场景	160
7.1.2	学习引导	161
7.1.3	相关知识与技术	161
7.1.4	任务的设计与实现	164
7.2	任务 2: 利用角色授权技术实施应用程序功能的代码级保护	166
7.2.1	任务场景	166
7.2.2	学习引导	167

7.2.3	相关知识与技术	167
7.2.4	任务的设计实现	170
7.3	课业任务及指导	174
学习单元 8	使用 ASP.NET Ajax 技术优化 ASP.NET 应用程序	176
8.1	任务 1: 使用 UpdatePanel 实现页面的局部更新	176
8.1.1	任务场景	176
8.1.2	学习引导	177
8.1.3	相关知识与技术	177
8.1.4	任务的设计与实现	182
8.2	任务 2: 使用 UpdateProgress 实现登录等待	189
8.2.1	任务场景	189
8.2.2	学习引导	189
8.2.3	相关知识与技术	190
8.2.4	任务的设计实现	191
8.3	任务 3: 使用 Timer 实现页面定时刷新	194
8.3.1	任务场景	194
8.3.2	学习引导	194
8.3.3	相关知识与技术	195
8.3.4	任务的设计与实现	196
8.4	课业任务及指导	201
学习单元 9	综合项目训练案例——E-learning 多媒体学习社区的设计与实现	204
9.1	项目背景及需求分析	204
9.1.1	项目背景	204
9.1.2	需求分析	204
9.2	系统总体设计	205
9.2.1	系统架构设计	205
9.2.2	系统数据库设计	205
9.2.3	系统的文件及其组织结构	208
9.3	功能的设计与实现	208
9.3.1	登录注册功能模块设计与实现	208
9.3.2	页面导航功能设计与实现	216
9.3.3	学习社区前台功能模块设计与实现	216
9.3.4	学习社区后台资源管理功能模块设计与实现	232
9.3.5	学习社区后台系统管理功能模块设计与实现	233

学习单元 1 认识 ASP.NET 并设计一个 简单的 ASP.NET Web 应用程序

由于 B/S 模式应用程序具有异地浏览和异地信息采集的特性，只要可以使用浏览器上网，就可以任何时间、任何地点、任何系统访问并运行软件。随着互联网的普及，B/S 模式软件得到广泛应用。尽管还有其他 Web 应用程序架构，但是由于微软的 ASP.NET 功能强大、易入门、易开发、所见即所得、开发周期短、运行效率高，ASP.NET 已经成为当前应用最广泛的 Web 应用程序架构之一。

ASP.NET 是什么？ASP.NET Web 应用程序又是怎么创建的？让我们行动起来，获取必要的知识和技能，创建一个简单 ASP.NET Web 应用程序，然后将它运行起来。

1.1 认识.NET Framework

如图 1-1 所示，ASP.NET Web 应用程序运行在 .NET Framework 基础上。不同版本的 ASP.NET 需要相应版本的 .NET Framework 支持，或者说不同版本的 ASP.NET Web 应用程序必须在安装相应版本的 .NET Framework 后才可以运行。本书介绍的 ASP.NET 3.5，相应地需要 .NET Framework 3.5 的支持。

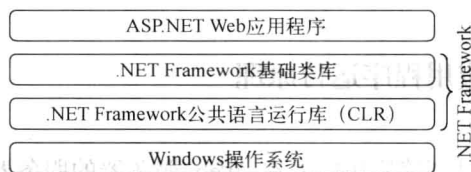


图 1-1 .NET Framework 的结构

1.1.1 认识.NET Framework 的组成

从图 1-1 可以看出，.NET Framework 建立在操作系统之上，主要由公共语言运行库和基础类库组成。

1. 公共语言运行库

与传统的应用程序不同，.NET 应用程序不直接运行在操作系统之上，而是运行在 .NET Framework 构建的“托管”运行环境上，此“托管”运行环境称为公共语言运行库（Common Language RunTime，简称 CLR）。因此，我们可以通俗地理解 .NET Framework 起到了类似于操作系统的作用。

.NET 应用程序编译后的代码不是机器代码，而是一种叫做 MSIL（Microsoft Intermediate Language，微软中间语言）的中间语言，当 .NET 应用程序运行时，需要一种即

时编译器将 MSIL 中间语言程序编译为机器语言，然后才可运行该程序。这种即时编译器就是公共语言运行库提供的。

2. .NET Framework 基础类库

如果说公共语言运行库为 .NET 应用程序提供了运行的环境，那么 .NET Framework 基础类库则为 .NET 应用程序提供了全部的基本功能，或者说包括 ASP.NET Web 应用程序在内的所有的 .NET 应用程序的所有功能都是通过调用 .NET Framework 基础类库中的类来实现的。成功安装 .NET Framework 后，.NET Framework 基础类库也随之安装。

1.1.2 选择合适的 .NET 语言

前面说过，不论是什么语言开发的 .NET 应用程序的源程序，其编译后的程序代码都是一种叫做 MSIL 的与程序设计语言和 CPU 指令无关的中间语言，等到程序运行时再由即时编译器编译为 CPU 指令相关的机器代码。这就意味着不论用什么语言开发的程序代码，只要将其编译为 MSIL，其他任何符合微软公共语言规范的程序设计语言（即能够将程序源代码编译为 MSIL 的程序设计语言）都可以调用该代码。也就是说可以用任何符合微软公共语言规范的程序设计语言开发 .NET 应用程序，甚至一个应用程序项目可以用多种语言混合开发。

目前流行的 .NET 应用程序设计语言为 Visual C#.NET 和 Visual Basic.NET。C# 是 .NET 中引入的全新语言，它与当前流行的 C、C++、Java 等程序设计语言的基本语法非常类似，是首选的 .NET 应用程序设计语言。

1.2 认识、创建并运行 ASP.NET Web 应用程序

1.2.1 ASP.NET Web 应用程序运行原理

1. 认识 ASP.NET

Web 应用程序是指通过浏览器访问并在 Web 服务器的服务器端运行、运行的结果又在浏览器中显示输出的应用程序。

应用程序可分为 C/S（客户端/服务器端程序）和 B/S（浏览器/服务器端程序）两种运行模式。C/S 应用程序由客户端应用程序和服务器端应用程序组成，客户端应用程序负责用户输入和输出，应用程序的运行逻辑则由服务器端完成。对于 C/S 应用程序，特定的服务器端应用程序必须有相应的客户端应用程序访问。Web 应用程序是 B/S 模式应用程序，其服务器端应用程序均可以使用浏览器通过 HTTP 协议进行访问，其输入输出都在浏览器中，因而不需要额外的客户端应用程序。同样是 Web 应用程序，不同的公司采用不同的具体解决方案，因此形成不同的编程架构方案，如 ASP、ASP.NET、PHP 和 JSP 等，ASP.NET 只是其中一种由微软开发的、取代早期版本 ASP 的 Web 应用程序编程架构解决方案。从如图 1-2 所示 Web 应用程序运行原理可知，ASP、ASP.NET、PHP 和 JSP 的区别实质上就是服务器端处理程序不同而已。

同样是 HTTP Web 页面请求，Web 服务器是通过文件的扩展名识别请求类别的。如果扩展名为 .htm 或者 .html，Web 服务器找到文件后并不交给服务器端处理程序处理，而是将其直

接发送给浏览器，故此类网页叫做静态网页；如果扩展名为 .asp、.aspx、.jsp 和 .php，Web 服务器接到请求后则将 Web 页面分别交由 ASP、ASP.NET、JSP 或 PHP 等相应的服务器端处理程序进行处理，生成浏览器可解释执行的 HTML 代码和 JavaScript 脚本后再发送给浏览器。

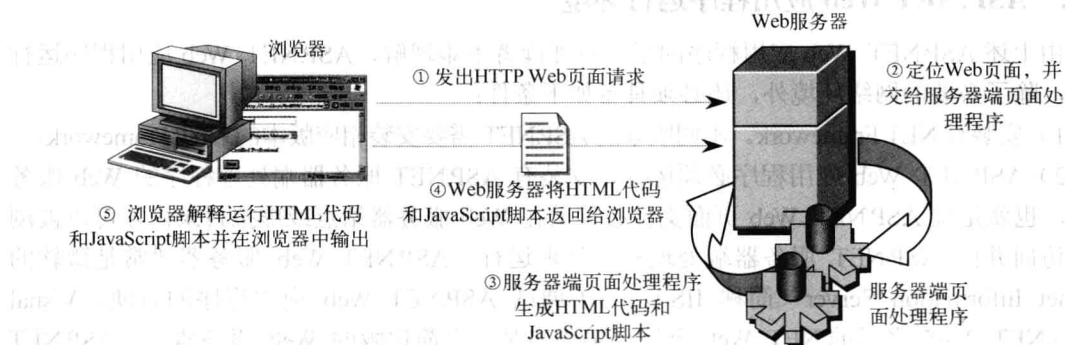


图 1-2 Web 应用程序运行原理示意图

2. ASP.NET Web 应用程序运行原理

ASP.NET 应用程序运行原理如图 1-3 所示。

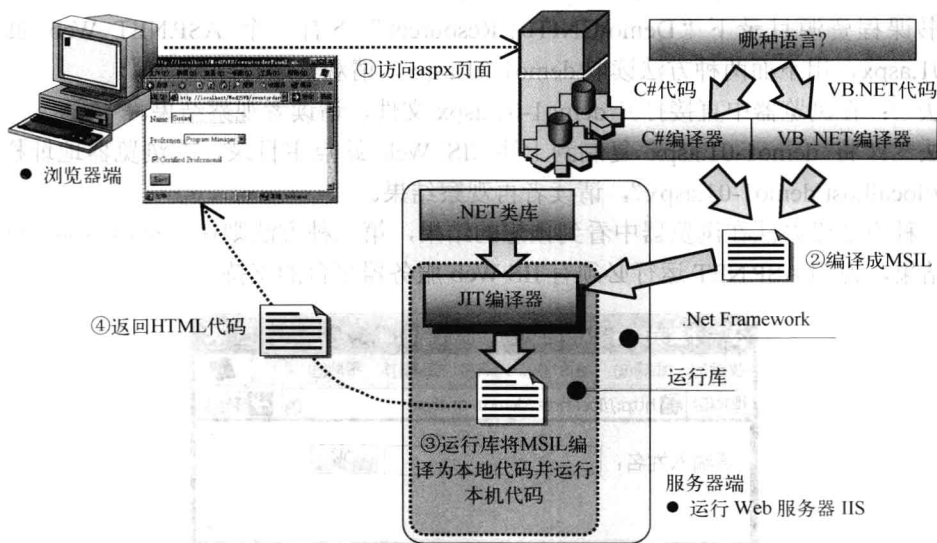


图 1-3 ASP.NET Web 应用程序运行原理示意图

ASP.NET 可以概括地理解为运行在 IIS 服务器上，以 .NET Framework 为服务器端页面处理平台的 Web 应用程序架构。

图中各步如下。

第①步：用户在客户端输入请求的 aspx 页面的 URL 地址及相关参数，向 Web 服务器发出 HTTP 请求。

第②步：如果该页面是首次访问，则 ASP.NET 根据程序设计语言选择相应的编译器将程序源代码编译为 MSIL，并以 dll 文件形式保存在站点中。如果不是首次访问，则跳过本步骤。

第③步：.NET 即时编译器 JIT 将 MSIL 编译为本机机器代码，同时运行生成 HTML 代

码和 JavaScript 脚本。

第④—⑤步：Web 服务器将生成的 HTML 和 JavaScript 脚本返回给浏览器显示。

1.2.2 ASP.NET Web 应用程序运行环境

由上述 ASP.NET Web 应用程序的运行原理读者不难理解，ASP.NET Web 应用程序运行环境，除了必要的网络环境外，还必须具备如下条件：

1) 安装有 .NET Framework。不同版本的 ASP.NET 需要安装相应版本的 .NET Framework。

2) ASP.NET Web 应用程序必须运行在安装有 ASP.NET 服务器端处理程序的 Web 服务器上，也就是说 ASP.NET Web 页面文件必须放在 Web 服务器站点或虚拟目录内才可以被浏览器访问并由 ASP.NET 服务器端处理程序处理运行。ASP.NET Web 服务器通常是微软的 Internet Information Server（简称 IIS）。为了调试 ASP.NET Web 应用程序的方便，Visual Studio.NET 2005 之后的 .NET Web 开发环境中自带一个简化版的 Web 服务器——ASP.NET Development Server，若在开发环境中运行 ASP.NET Web 应用程序，该 Web 服务器将自动启动并运行该 ASP.NET Web 应用程序。但是 ASP.NET Development Server 毕竟是一个简化版的 Web 服务器，没有管理上、安全上的健壮性，所以应用程序开发完成后，应该部署到 IIS 等支持 ASP.NET Web 应用程序的专业 Web 服务器平台上。

本书课程资源目录下“Demo\UNIT01\Resource\”下有一个 ASP.NET Web 页面文件 demo1-01.aspx，用下面两种方法读取 demo1-01.aspx，请观察运行结果。

方法一：在浏览器中直接打开 demo1-01.aspx 文件，请读者观察结果。

方法二：将 demo1-01.aspx 复制到上述 IIS Web 站点主目录，在浏览器地址栏中输入“HTTP://localhost/demo1-01.aspx”，请读者再观察结果。

第一种方法没办法在浏览器中看到预想的结果，第二种方法则在浏览器中显示如图 1-4 所示的结果，说明 ASP.NET 运行必须有 IIS Web 服务器平台的支持。



图 1-4 一个简单的 ASP.NET Web 页面

1.2.3 ASP.NET Web 应用程序构成

1. ASP.NET Web 应用程序构成

如图 1-5 所示，ASP.NET Web 应用程序由 Web 窗体.aspx 文件及其代码隐藏页.aspx.cs 文件、全局应用程序类 Global.asax 文件、配置文件和程序集文件构成。

1) aspx 文件及其代码隐藏页：用户输入、输出页面文件及其运行程序代码文件。

2) 全局应用程序类 Global.asax 文件：包含响应应用程序级别事件的代码，每个

ASP.NET 应用程序 Global.asax 是唯一的，且必须驻留在该应用程序（站点）的根目录中。

3) 应用程序配置文件：设置 Web 应用程序运行方式及运行环境。

4) 程序集文件：Web 窗体.aspx 文件及其代码隐藏页.aspx.cs 文件一旦编译发布后，就以 dll 文件保存在/bin 目录下，原来的.cs 文件不再有意义。将此 dll 文件叫做程序集。



图 1-5 ASP.NET Web 应用程序组成

2. ASP.NET Web 应用程序目录结构

如图 1-6 所示，应用程序所有文件必须在站点主目录（站点根目录）及其子目录下，其中，应用程序编译后生成的 MSIL 程序集必须在/bin 目录下，Global.asax 必须在站点根目录下，Web.config 则可在任何目录下（只是作用范围不同。具体地说，其对当前目录及其子目录程序有效），代码隐藏页文件在其相应的 aspx 和 asmx 同级目录中。代码隐藏页代码编译后将生成程序集文件放在/bin 目录下，所以应用程序发布后代码隐藏页就没必要了。图 1-6 所示的是程序发布后的情况，所以代码隐藏页文件没有列出。

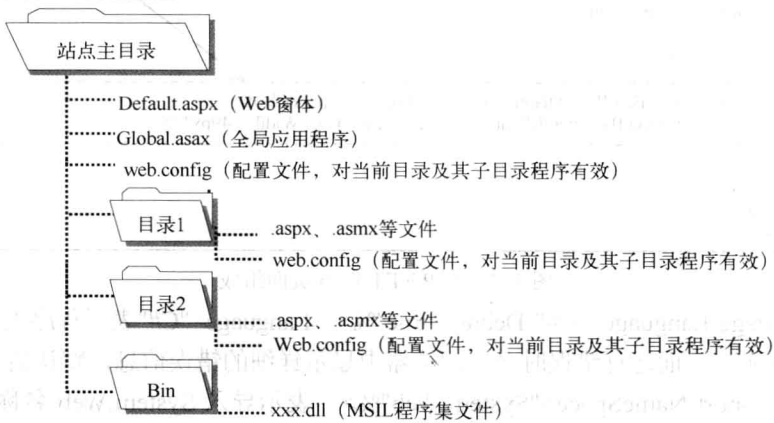


图 1-6 ASP.NET 应用程序目录结构

1.3 认识 ASP.NET Web 页面

ASP.NET Web 应用程序的功能主要是通过访问和运行.aspx 页面实现的。下面简要介绍.aspx 页面的组成及设计方法。

1.3.1 ASP.NET Web 页面组成

类似于 HTML 网页，ASP.NET Web 页面由标记语言和程序代码构成，所不同的是前者

程序代码是浏览器端运行的，而 ASP.NET Web 页面是服务器端运行的，运行的结果再返回给浏览器端运行。

打开本书课程资源的“Demo\UNIT01\Resource\”目录下的 ASP.NET Web 页面文件 demo1-01.aspx，可以看到如图所示的页面的源代码。

1. 指令

指令用于指示编译程序如何编译该 ASP.NET Web 页面，一般放在页面代码的顶部，其格式为：

```
<%@指令名 属性名 1="属性值 1" 属性名 2="属性值 2" .....%>
```

常见的指令有 page 页面指令和 Import 导入指令。图 1-7 中的两条指令分别解释如下：

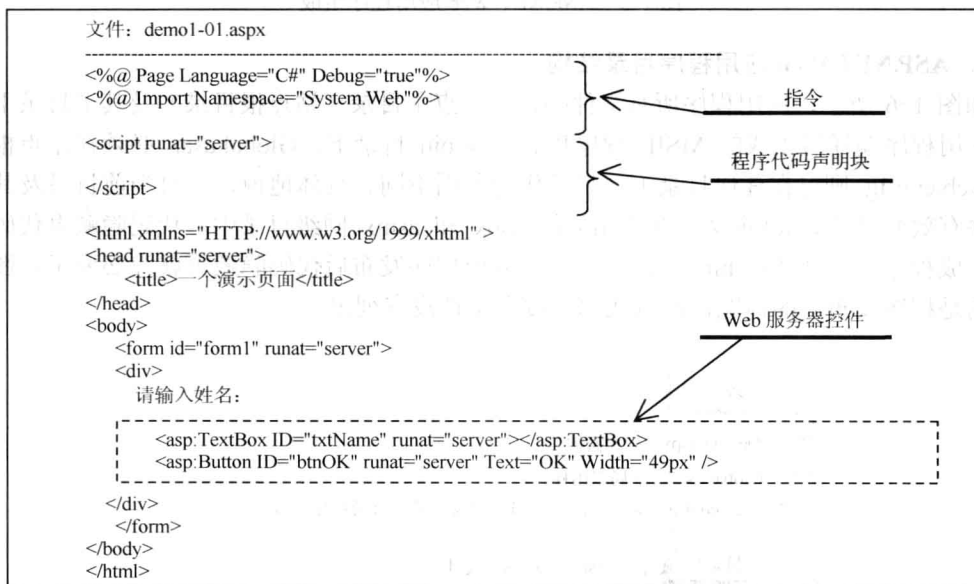


图 1-7 ASP.NET Web 页面组成

1) `<%@ Page Language="C#" Debug="true"%>` `Language="C#"`表示程序设计语言为 C#；`Debug="true"`表示当页面运行错误时会在浏览器中显示详细的错误信息，默认值为 `false`。

2) `<%@ Import Namespace="System.Web"%>` 表示导入 `System.Web` 名称空间。名称空间可以在页面中导入，也可以在扩展名为 `.cs` 的代码文件中导入。如果在 `.cs` 的代码文件中导入，则用 `using` 命令，如果在页面中导入则用 `Import` 指令，千万不要搞混淆。一般情况下，使用 .NET 中的类须事先导入命名空间，但是为了程序员方便，标准的 ASP.NET 名称空间默认已经导入，所以导入指令可以省略。

2. 代码声明块

代码声明块放在 `<Script runat="server"></Script>` 标记中，用于定义页面后台运行的程序代码（如后面将讨论的事件函数）。要注意与 JavaScript 脚本的 `<Script>` 标记的区别。代码声明块所在的 `<Script>` 标记具有 `runat="Server"` 属性，指示该标记是服务器端的标记。

3. ASP.NET Web 控件

Web 控件是生成页面的界面元素。使用时，Web 控件必须放在服务器端标记 `<form`

runat="server">中。ASP.NET Web 控件格式为:

```
<asp:控件名 id="控件名称" runat="server"属性 1="值 1" 属性 2="值 2" .....  
事件名="事件函数名"></asp:控件名 >
```

标准的 ASP.NET Web 控件标记必须以“<asp:控件名.....>”开头,以“</asp:控件名>”结束。图 1-7 所示的代码中的<asp:TextBox ID="txtName" runat="server"></asp:TextBox>和<asp:Button ID="btnOK" runat="server" Text="OK" Width="49px" />两个标记即为两个 ASP.NET Web 控件标记,控件名 TextBox 表示输入文本的文本框,控件名 Button 为一个标题为“OK”的按钮。

4. HTML 标记

可以认为 ASP.NET 页面的基本结构是在 HTML 基本结构的基础上加上了指令、声明、代码声明块和 ASP.NET Web 控件。另外,尽管页面中所有的具体内容都可以用 ASP.NET Web 控件生成,但是在实际应用中,如果页面中内容是静态的,一般直接用 HTML 标记。

1.3.2 使用 HTML 标记和 Web 控件设计页面

ASP.NET 语言程序设计的基本方法是先使用 HTML 和 Web 控件设计页面界面,然后编写相应的应用程序代码。编写程序代码方法将在下一节讨论。

使用 HTML 和 Web 控件设计页面界面的基本步骤如下:

- 1) 新建加入或打开 Web 页面。
- 2) 通过 HTML 设计页面的静态内容。
- 3) 根据需要在页面适当的位置加入 Web 控件。
- 4) 设置 Web 控件相关属性。
- 5) 如果有多个控件则重复 3) 和 4)。

下面以实现图 1-8 所示 ASP.NET Web 登录界面为实例进一步讨论使用 HTML 和 Web 控件设计页面界面的方法。

1) 准备页面的图片资源。因为本书学习单元 1 课程资源目录“Demo\UNIT01\Resource\images\”内已保存有图 1-8 所示 ASP.NET Web 登录界面所需要的所有图片,所以,将“Demo\UNIT01\Resource\”下的 images 目录直接复制到 demo1-02.aspx 所在的目录中,即可完成页面的图片资源准备工作。

2) 在 Visual Studio.NET 中新建一个名为 demo1-02.aspx 的 ASP.NET 页面。

3) 设计页面静态内容。本实例 HTML 静态内容笔者已经做好,放在本书课程资源的“Demo\UNIT01\Resource\”目录下的 login.txt 文件中。切换到 demo1-02.aspx 的代码视图,然后在<body></body>标记内加入服务器端表单<form id="form1" runat="server"></form>标记,最后将 login.txt 文件的内容复制到上述服务器端表单之间,即可完成如图 1-9 所示的页面静态内容的设计(图中空白处为表格的单元格)。

4) 设计页面的 Web 控件。具体设计方法为:在如图 1-9 所示的页面 4 个空白单元格对应位置的<td></td>标记间分别加入两个 TextBox 文本框控件和两个 ImageButton 图像按钮控件。前者分别用于用户输入用户名和密码,后者用于根据用户是管理员或一般用户分别采用不同的按钮进行登录。

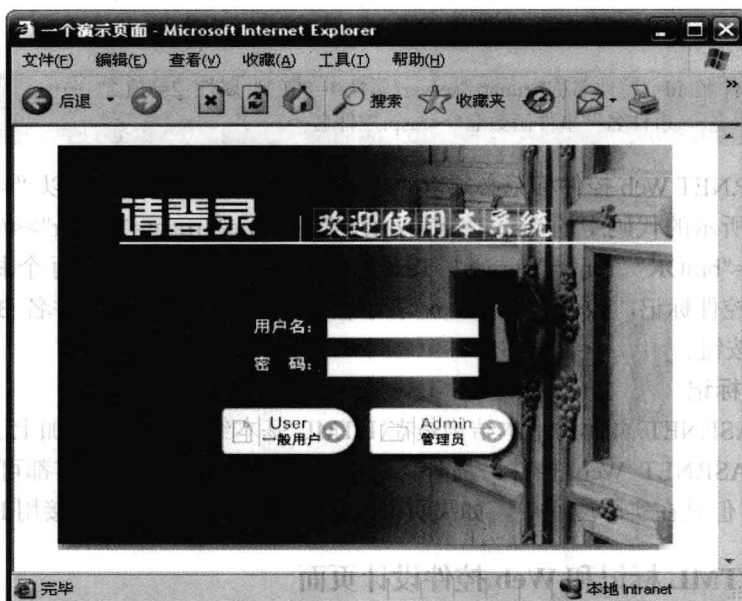


图 1-8 一个 ASP.NET Web 登录界面示例

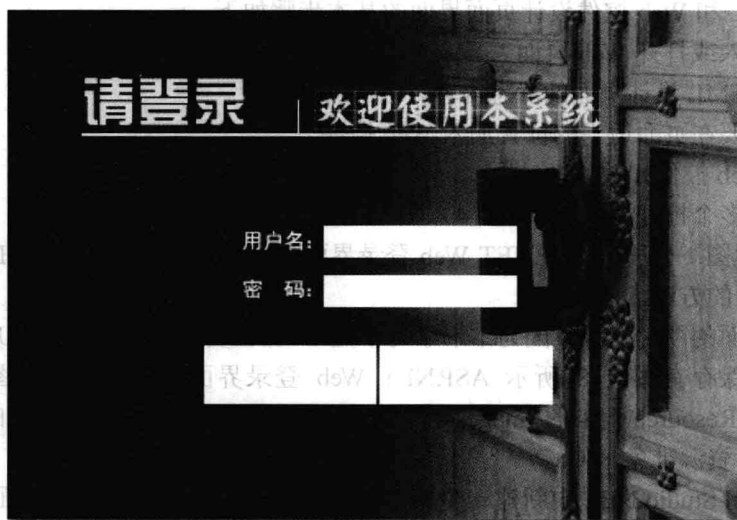


图 1-9 demo1-02.aspx 静态界面效果

4 个控件的代码如下。

1) 用户名文本框:

```
<asp:TextBox ID="txtName" runat="server" Height="12px" Width="110px" TextMode="SingleLine"/>
```

2) 密码文本框:

```
<asp:TextBox ID="txtPwd" runat="server" Height="12px" Width="110px" TextMode="Password"/>
```

3) 用户登录图像按钮:

```
<asp:ImageButton ID=" btnUser " runat="server" Height="37px" Width="105px" ImageUrl="images/login01-3_13.jpg"/>
```

4) 管理员登录图像按钮:

```
<asp:ImageButton ID=" btnAdmin" runat="server" Height="37px" ImageUrl="images/login01-3_15.jpg" Width="106px" />
```

至此, 图 1-8 所示登录界面实例全部完成。请读者将 demo1-02.aspx 连同其相关的 images 目录复制到上述 IIS Web 站点主目录, 然后在浏览器中通过 HTTP 协议访问 demo1-02.aspx, 即可验证设计结果。

1.4 在 Web 应用程序中添加程序代码——运行逻辑

1.4.1 事件驱动程序设计原理

应用程序启动后即进入等待状态, 当系统或用户在程序界面上进行一个操作 (如双击鼠标、单击鼠标、右击鼠标、拖放鼠标等), 主程序就会调用相应的处理程序响应, 完成相应的任务。上述操作称为事件, 对应的处理程序称为事件处理程序。对此, 在程序设计过程中, 程序员所做的主要工作是决定应用程序响应什么事件, 并编写该事件的事件处理程序, 我们将此种程序设计方式称为事件驱动程序设计。

在 C# 中, 事件处理程序以函数的形式供主程序调用, 因此, 事件处理程序通常称为事件函数 (如图 1-10 所示)。

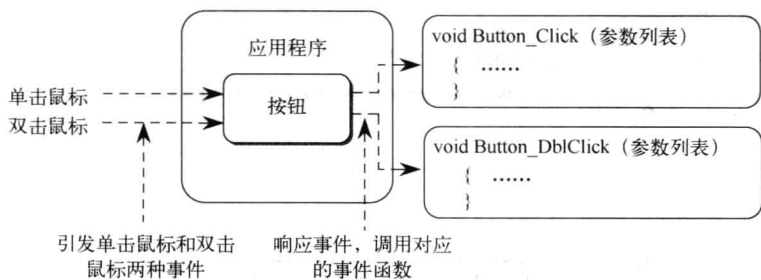


图 1-10 事件驱动程序运行示意图

1.4.2 响应页面及控件事件

ASP.NET 支持事件驱动程序设计。设计好 ASP.NET Web 页面的界面后, 就可以编写事件函数响应发生在页面和 Web 控件上可支持的事件 (所谓可支持的事件是指页面、控件设计者事先预设的页面、控件能够“感知”的操作)。不同的控件能够响应的事件是不同的, 下面仅对页面和控件事件响应的一般方法做一简单介绍, 并且对页面的常用事件做一些说明。至于控件的具体事件, 将在后续单元具体说明。