

ASP.NET 4.0 WANGZHAN JIANSHE JICHU JIAOCHENG



世纪高职高专规划教材
高等职业教育规划教材编委会专家审定

ASP.NET 4.0 网站建设基础教程

主 编 王 翔
副主编 刘丽丽 陈 薇



北京邮电大学出版社
www.buptpress.com



21 世纪高职高专规划教材

高等职业教育规划教材编委会专家审定

ASP.NET 4.0 网站建设基础教程

主 编 王 翔

副主编 刘丽丽 陈 薇



北京邮电大学出版社
www.buptpress.com

内 容 简 介

本书以一个基于三层架构设计与开发的“企业新闻发布网站”项目作为实践主线,为突出 ASP.NET 的实践特点和应用方向,采用了“先学习、后操作,先模仿、后思考”的模式,将 Web 应用程序设计与开发过程中所必须掌握的知识归纳为若干案例,每个案例解决一个问题。初学者可在简要学习案例涉及知识点后,模仿案例,获得直接体验,然后再次回顾和案例相关的知识,在实操中加深理解并巩固学习成果。通过案例的逐一串联,逐步地构成完整的知识体系,有利于初学者快速掌握 ASP.NET 应用程序设计中最常用、最核心的知识和技能。全书共分 9 章,内容包括走进 ASP.NET4.0、C#4.0 语言基础、面向对象程序设计基础、使用服务器控件、内置对象、数据库与数据访问控件、ADO.NET 访问数据库、创建统一风格的网站等。

本书立足于让学生能“看得懂、学得会、用得上”,强调学生技能的培养,是一本面向高职高专院校计算机应用、计算机软件、计算机网络、电子商务等专业的实用型能力教材。此外,本书也可以作为网站开发与建设技术的培训教材和自学读本。

图书在版编目(CIP)数据

ASP.NET 4.0 网站建设基础教程/王翔主编. --北京:北京邮电大学出版社,2012.6

ISBN 978-7-5635-2960-5

I. ①A… II. ①王… 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(2012)第 058840 号

书 名: ASP.NET 4.0 网站建设基础教程
主 编: 王 翔
责任编辑: 彭 楠
出版发行: 北京邮电大学出版社
社 址: 北京市海淀区西土城路 10 号(邮编:100876)
发 行 部: 电话: 010-62282185 传真: 010-62283578
E-mail: publish@bupt.edu.cn
经 销: 各地新华书店
印 刷: 北京源海印刷有限责任公司
开 本: 787 mm×1 092 mm 1/16
印 张: 15.25
字 数: 380 千字
印 数: 1—3 000 册
版 次: 2012 年 6 月第 1 版 2012 年 6 月第 1 次印刷

ISBN 978-7-5635-2960-5

定 价: 32.00 元

• 如有印装质量问题,请与北京邮电大学出版社发行部联系 •

前 言

.NET 技术是微软近年来推出的主要技术,自从 .NET 2.0 版本问世之后,越来越多的开发人员和企业已经能够接受 .NET 技术带来的革新。ASP.NET 4.0 是目前微软最新的 Web 应用开发可视化平台,它不仅在语言和技术上弥补了原有的 ASP.NET 2.0 与 ASP.NET 3.5 的不足,并提供了很多新的控件和技术特性以提升开发人员的生产力。与之相应,Visual Studio 2010 除了保持与 Visual Studio 旧版本相同的特点之外,也提供了大量新的特色帮助提升开发人员的编程效率。

本书站在实践与应用的角度,比较系统地介绍了 ASP.NET 4.0 及其应用与开发技术,主要包括走进 ASP.NET 4.0、C# 4.0 语言基础、面向对象程序设计基础、使用服务器控件、内置对象、数据库与数据访问控件、ADO.NET 访问数据库、创建统一风格的网站及三层架构系统的搭建等内容。

鉴于 ASP.NET 所涉及的内容众多,本书尽可能介绍各个方面的主要内容,对实际设计与开发过程中使用较少的知识点只作了简略介绍,而对那些应用性强、开发中使用频率较高的知识点则通过一系列小规模精选案例进行了相对全面、深入的阐述和分析。

本书第 1、4、6、7、9 章由王翔编写,第 2、3 章由刘丽丽编写,第 5、8 章由陈薇编写,沈明、陈忠、傅宜宁、李莉等老师参与了部分内容的编写。本书的编写得到了广州航海高等专科学校及广东第二师范学院的大力支持与帮助,再此一并表示感谢。

限于作者的水平,书中难免有不妥、疏漏之处,敬请广大读者批评指正。

编 者

目 录

第 1 章 走进 ASP.NET 4.0	1
1.1 Web 服务器工作流程	1
1.2 Web 开发技术的发展历程	2
1.2.1 静态网页与动态网页	2
1.2.2 动态网页技术	2
1.3 .NET 与 ASP.NET	3
1.3.1 什么是 .NET	3
1.3.2 .NET 应用程序框架	4
1.3.3 ASP.NET	5
1.4 初识 Microsoft Visual Studio 2010	6
1.4.1 安装 Visual Studio 2010	6
1.4.2 Visual Studio 2010 集成开发环境	7
1.5 第一个 ASP.NET 应用程序实例	10
1.5.1 创建 Web 应用程序的一般步骤	10
1.5.2 一个简单的 ASP.NET 应用程序实例	11
习题	14
第 2 章 C# 4.0 语言基础	16
2.1 ASP.NET 中的 C# 程序	16
2.1.1 在 Web 网站中创建 C# 程序	16
2.1.2 代码隐藏页模型	18
2.1.3 C# 程序的结构	19
2.2 数据类型	20
2.2.1 值类型	20
2.2.2 引用类型	21
2.2.3 装箱与拆箱	22
2.3 常量与变量	23
2.3.1 常量	23
2.3.2 变量	24
2.4 运算符与表达式	25
2.4.1 算术运算符	25
2.4.2 关系运算符	25
2.4.3 逻辑运算符	26
2.4.4 位运算符	26
2.4.5 赋值运算符	27
2.4.6 条件运算符	27
2.4.7 自增和自减运算符	27

2.4.8 字符串连接运算符	28
2.4.9 运算符的优先级与结合性	28
2.5 流程控制	29
2.5.1 条件语句	29
2.5.2 循环结构	33
2.5.3 跳转结构	36
2.6 数组、结构与枚举	38
2.6.1 数组	38
2.6.2 结构	43
2.6.3 枚举	44
习题	46
第3章 面向对象程序设计基础	48
3.1 什么是面向对象	48
3.1.1 面向过程程序设计的概念	48
3.1.2 面向对象程序设计的概念	49
3.2 类与对象	49
3.2.1 类	50
3.2.2 对象	51
3.2.3 构造函数和析构函数	53
3.3 属性	56
3.3.1 get 访问器	56
3.3.2 set 访问器	57
3.4 方法	57
3.4.1 方法的声明	57
3.4.2 方法的参数传递	58
3.4.3 方法的重载	61
3.5 类的继承性与多态性	62
3.5.1 继承性	62
3.5.2 多态性	64
3.6 命名空间	68
3.6.1 命名空间的定义与使用	68
3.6.2 .NET Framework 中的命名空间	69
习题	70
第4章 使用服务器控件	72
4.1 服务器控件概述	72
4.1.1 服务器控件的类型	72
4.1.2 服务器控件的类层次结构	73
4.2 ASP.NET 页面与控件的生命周期	73
4.3 HTML 服务器控件	77
4.3.1 HTML 服务器控件的层次结构	77
4.3.2 HTML 服务器控件的使用概要	77
4.3.3 HTML 服务器控件的综合应用	79
4.4 标准服务器控件	83

4.4.1	Web 服务器控件的层次结构	83
4.4.2	Web 服务器控件的使用概要	84
4.4.3	数据显示与输入控件	86
4.4.4	数据提交控件	89
4.4.5	选择和列表控件	95
4.4.6	图像显示控件	102
4.4.7	广告轮显控件(AdRotator)	104
4.4.8	日历控件(Calendar)	106
4.4.9	文件上传控件(FileUpload)	107
4.4.10	面板控件(Panel)	109
4.5	验证控件	111
4.5.1	验证控件概述	111
4.5.2	验证控件共有属性	112
4.5.3	必填字段验证控件(RequiredFieldValidator)	112
4.5.4	比较验证控件(CompareValidator)	113
4.5.5	范围验证控件(RangeValidator)	114
4.5.6	正则表达式验证控件(RegularExpressionValidator)	115
4.5.7	自定义验证控件(CustomValidator)	118
4.5.8	验证错误摘要控件(ValidationSummary)	119
习题		120
第 5 章	内置对象	123
5.1	Page 类与 Page 对象	123
5.2	Response 对象	124
5.3	Request 对象	125
5.4	Server 对象	129
5.5	Session 对象	132
5.6	Application 对象	134
习题		136
第 6 章	数据库与数据访问控件	138
6.1	数据库基础	138
6.1.1	数据库管理系统	138
6.1.2	表和视图	138
6.1.3	SQL 语言	139
6.1.4	操作 SQL Server 2008 Express 数据库	141
6.2	SqlDataSource 数据源控件	142
6.3	数据绑定控件	146
6.3.1	GridView 控件	146
6.3.2	DetailsView 控件	153
6.3.3	FormView 控件	156
6.3.4	Repeater 控件与内部数据绑定	157
习题		161
第 7 章	ADO.NET 访问数据库	162
7.1	ADO.NET 概述	162

7.2 初识 ADO.NET 核心组件	163
7.3 Connection 对象	164
7.4 Command 对象	165
7.5 DataReader 对象	166
7.6 DataAdapter 对象	169
7.6.1 DataAdapter 对象的常用属性	169
7.6.2 DataAdapter 对象的常用方法	169
7.7 DataSet 对象	170
7.7.1 数据表和数据表集合	170
7.7.2 数据列和数据列集合	172
7.7.3 数据行和数据行集合	173
7.7.4 DataSet 对象的属性和方法	174
习题	180
第8章 创建统一风格的网站	181
8.1 主题和皮肤	181
8.1.1 CSS 基础	181
8.1.2 皮肤文件	184
8.1.3 主题	185
8.2 母版页与内容页	192
8.2.1 母版页的创建与代码结构	192
8.2.2 内容页的创建与代码结构	193
8.3 站点导航	196
8.3.1 站点地图与 SiteMapPath 控件	196
8.3.2 TreeView 控件	198
8.3.3 Menu 控件	200
习题	204
第9章 项目实践之三层架构新闻发布系统	205
9.1 三层架构概述	205
9.1.1 二层架构及其局限性	205
9.1.2 什么是三层架构	205
9.1.3 三层架构的优越性	206
9.1.4 三层架构的搭建实例	206
9.2 系统需求分析与功能设计	210
9.2.1 系统功能与流程设计	210
9.2.2 数据库设计	211
9.2.3 Model 层(实体层)的设计	213
9.2.4 DAL 层(数据访问层)的方法设计	215
9.2.5 BLL 层(业务逻辑层)的方法设计	221
9.3 UI 层(界面层或表示层)的设计	222
9.3.1 前台界面的设计与实现	222
9.3.2 后台界面的设计与实现	231
习题	235
参考文献	236

第 1 章 走进 ASP.NET 4.0

ASP.NET 是 Web 开发技术高速发展的产物,是当前主流的 Web 应用程序开发技术之一,它构建于 .NET Framework 之上,使得从传统的数据库访问技术到如今的分布式应用开发技术等一系列技术都发生了变革。而且,它在快速开发、编译与部署等方面的优势是任何一种互联网开发技术都无法比拟的。也正是因为这些优点,一批又一批的开发人员加入到 ASP.NET 开发阵营当中。通过 ASP.NET,我们可以简单快速地开发出企业级的、高性能的、便于维护的 Web 应用系统。

为了让大家能够有一个良好的学习开端,对 Web 开发中的一些基础知识及 ASP.NET 的基本概念与技术特点等有一个较为清楚的认识,本章将从 Web 服务器的基本工作流程、Web 开发技术的发展历程入手,逐一介绍一些与 ASP.NET 密切相关的概念性知识,如什么是 .NET(读做“dot-net”)、什么是 .NET FRAMEWORK 等。除了这些必须有所了解的概念性知识外,本章还会对 Microsoft Visual Studio 2010 集成开发环境的构成及使用加以介绍,最后还将带领大家接触一个简单的 ASP.NET 应用程序案例。

1.1 Web 服务器工作流程

一个 Web 服务器通常也被称为 HTTP 服务器,它通过 HTTP 协议与客户端通信。这个客户端通常指的就是 Web 浏览器。

Web 服务器的工作流程可以简单的归纳为:客户端连接服务器;客户端向服务器发送资源请求;服务器收到这个请求后,将为用户查找资源并在依据资源类型进行相应处理后向客户机发送应答响应,将结果返回给客户端浏览器;客户机与服务器断开。如图 1.1 所示。一个简单的事务处理事件就是这样实现的,需要注意的是,客户端与服务器的通信是无连接的,也就是当服务器发送了应答后就与客户机断开连接,等待下一次请求。

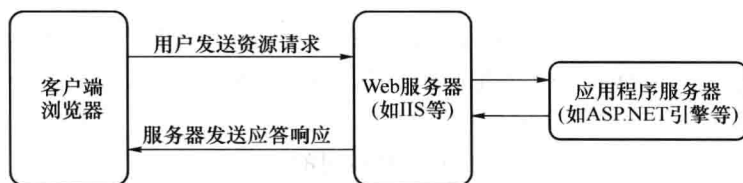


图 1.1 Web 服务器简化工作流程

服务器上的资源内容多种多样,既可以是普通的 HTML 页面、音频文件、视频文件或图片等,也可以是使用某种网页技术设计的动态页面。不同的资源类型对服务器的要求也各不相同。

1.2 Web 开发技术的发展历程

1.2.1 静态网页与动态网页

静态网页是早期的 Web 网站向用户提供信息的一种形式,它的扩展名通常为 htm 或 html。静态网页的内容一般由网站发布时即固定的文字、图片、视频等元素构成,这些内容只能由人工更新,网站的用户也只能被动地浏览网站提供的网页内容。其特点如下。

(1) 网页内容不会发生变化,除非设计者人工修改了网页的内容。

(2) 不能实现和网站用户之间的交互。信息流向是单向的,即只能从服务器到浏览器。服务器也不能根据用户的选择或需求调整返回给用户的内容。

随着网页技术应用的普及,人们对网页信息的及时更新提出了更高的要求。随着数据库和脚本技术(如 ASP、PHP 和 JSP 等)的发展,越来越多的站点开始采取新型的页面发布方式。在这一方式下,Web 服务器实现了与用户的交互,能够按照用户的需要动态地、即时地构造 Web 页面,这些动态构造的页面就被称为“动态网页”。通常动态网页都需要后台数据库的配合,从而实现留言板、论坛、聊天室、校友录等页面功能。

静态网页与动态网页的区别在于 Web 服务器对它们的处理方式不同。当 Web 服务器接收到对静态网页的请求时,服务器直接将该页发送给客户端浏览器进行解释,服务器不进行任何处理。如果接收到对动态网页的请求,在从 Web 服务器中找到该文件后,还需将它传递给一个被称做应用程序服务器的特殊模块,由它负责编译和执行网页,将执行后的结果传递给客户端浏览器。

静态网页是网站建设的基础,静态网页和动态网页之间也并不矛盾,动态网站建设时可以采用“动静结合”的原则,适合采用动态网页的地方用动态网页,如果没有交互需求也可以考虑用静态网页的方法来实现。另外,有时为了网站适应搜索引擎检索的需要,即使采用动态网站开发技术,也会将网页内容转化为静态网页发布。

1.2.2 动态网页技术

1. CGI

CGI(Common Gateway Interface,公共网关接口)是添加到 Web 服务器的模块,提供了在服务器上创建脚本的机制。CGI 允许用户调用 Web 服务器上的另一个程序(如 Perl 脚本)来创建动态 Web 页,且 CGI 的作用是将用户提供的数据传递给该程序进行处理,以创建动态 Web 应用程序。CGI 可以运行于很多不同的平台(如 UNIX 等)。不过 CGI 存在不易编写、消耗服务器资源较多的缺点。

2. JSP

JSP(Java Server Pages)是一种允许用户将 HTML 或 XML 标记与 Java 代码相组合,从而动态地生成 Web 页的技术。JSP 允许 Java 程序利用 Java 平台的 JavaBeans 和 Java 库,运行速度比 ASP 快,具有跨平台特性。已有允许用户在 IIS 服务器中使用 JSP 的插件

模块。

3. PHP

PHP 是指超文本预处理程序 (HyperText Processor)。它起源于个人主页 (Personal Home Pages), 使用一种创建动态 Web 页的脚本语言, 语法类似 C 语言和 Perl 语言。PHP 是开放源代码和跨平台的, 可以在 Windows NT 和 UNIX 上运行。PHP 的安装较复杂, 会话管理功能不足。

4. ASP

ASP (Active Server Pages, 动态服务器页面) 是 Web 服务器上的模块 (asp.dll 文件), 它允许使用 VBScript 和 JavaScript 脚本语言编程, 在服务器端使用 Windows 提供的任何功能, 如数据库存取、E-mail 收发、网络功能、文件处理、图形处理、系统功能等, 但它只能在 Windows 平台上运行。

5. ASP.NET

ASP.NET 是一种基于 .NET 框架开发动态网页的新技术, 它依赖于 Web 服务器上的 ASP.NET 模块 (aspnet_isapi.dll 文件), 但该模块本身并不处理所有工作, 它将一些工作传递给 .NET 框架进行处理。ASP.NET 允许使用多种面向对象语言编程, 如 VB.NET、C#、C++、Jscript.NET 和 J#.NET 语言等, 它也只能在 Windows 平台上运行。

1.3 .NET 与 ASP.NET

ASP.NET 是微软推出的 ASP 的下一代 Web 开发技术。ASP.NET 顾名思义是基于 .NET 平台而存在的, 在了解 ASP.NET 之前就需要首先了解什么是 .NET 技术, 并了解 .NET 技术的核心——.NET 应用程序框架。

1.3.1 什么是 .NET

.NET 技术是微软近几年推出的主要技术, 微软为 .NET 技术的推出可谓是不遗余力。2002 年微软发布了 .NET 技术的第一个版本, 但是由于系统维护和系统学习的原因, .NET 技术当时并没有广泛地被开发人员和企业所接受。而自从 .NET 2.0 版本之后, 越来越多的开发人员和企业已经能够接受 .NET 技术带来的革新。

对于 .NET, 可以这样认为: .NET 是微软公司要提供的一系列产品的总称。具体说来, .NET 由以下部分组成: .NET 战略、.NET Framework、.NET 企业服务器和 .NET 开发工具。

- .NET 战略是指把所有的设备通过 Internet 连接在一起, 并把所有的软件作为这个网络所提供的服务的想法。
- .NET Framework 是一个程序设计环境, 它提供了具体的服务和技术, 方便开发人员建立相应的应用程序。
- .NET 企业服务器是指如 SQL Server 之类的 .NET Framework 应用程序使用的服务器端产品。它们虽然不是由 .NET Framework 编写成的, 但是它们都支持 .NET。

- 为了能够在 .NET Framework 上进行程序开发,微软把 Visual Studio 进行升级,并把升级后的产品命名为 Visual Studio.NET。这就是 .NET 开发工具。

1.3.2 .NET 应用程序框架

.NET 应用程序框架是一个多语言组件开发和执行环境,无论开发人员使用的是 C# 作为编程语言还是使用 VB.NET 作为其开发语言,都能够基于 .NET 应用程序框架而运行。.NET 应用程序框架主要包括三个部分,分别为公共语言运行库(Common Language Runtime, CLR)、基类库和活动服务器页面(ASP.NET),其结构如图 1.2 所示。

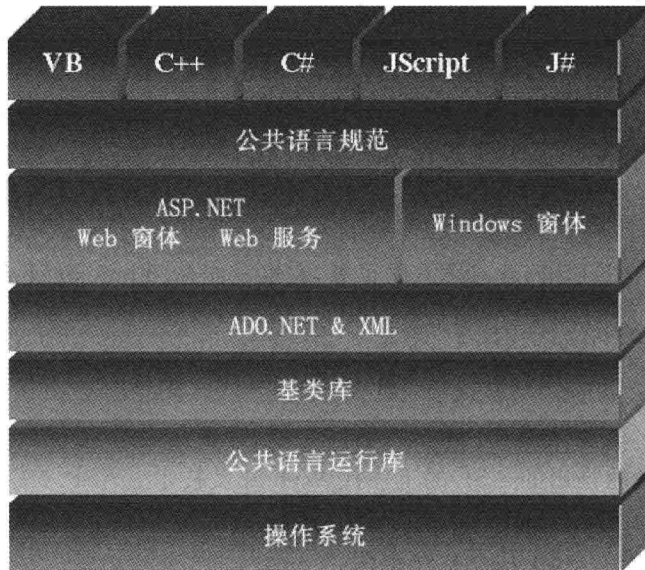


图 1.2 .NET 应用程序框架结构

1. 公共语言运行库

公共语言运行库在组件的开发及运行过程中扮演着非常重要的角色。在经历了传统的面向过程开发,开发人员寻找更多的高效的方法进行应用程序开发,这其中的发展成为了面向对象的应用程序开发,在面向对象程序开发的过程中衍生了组件开发。

在组件运行过程中,运行库负责管理内存分配、启动或删除线程和进程、实施安全性策略,同时满足当前组件对其他组件的需求。在多层开发和组件开发应用中,运行库负责管理组件与组件之间的功能的需求。

2. 基类库

.NET 框架为开发人员提供了一个统一、面向对象、层次化、可扩展的类库集(API)。现今,C++开发人员使用的是 Microsoft 基类库,Java 开发人员使用的是 Windows 基类库,而 Visual Basic 用户使用的又是 Visual Basic API 集,在应用程序开发中,很难将应用程序进行平台的移植,当出现了不同版本的 Windows 时,就会造成移植困难。

.NET 框架统一了微软当前的各种不同类型的框架。.NET 应用程序框架是一个系统级的框架,对现有的框架进行了封装,开发人员无须进行复杂的框架学习就能够轻松使用

.NET 应用程序框架进行应用程序开发。无论是使用 C# 编程语言,还是使用 Visual Basic 编程语言,都能够进行应用程序开发,不同的编程语言所调用的框架 API 都来自 .NET 应用程序框架,所以这些应用程序之间就不存在框架差异的问题,在不同版本的 Windows 中也能够方便地移植。



注意: .NET 框架能够安装到各个版本的 Windows 中,当有多个版本的 Windows 时,只要安装了 .NET 框架,任何 .NET 应用程序就能够在不同的 Windows 中运行而不需要额外移植。

3. 活动服务器页面

.NET 框架还为 Web 开发人员提供了基础保障,ASP.NET 是使用 .NET 应用程序框架提供的编程类库构建而成的,它提供了 Web 应用程序模型,该模型由一组控件和一个基本结构组成,使用该模型使 ASP.NET Web 开发变得非常容易。开发人员可以将特定的功能封装到控件中,然后通过控件的拖动进行应用程序的开发,这样不仅提高了应用程序开发的简便性,还极大精简了应用程序代码,让代码更具有复用性。

.NET 应用程序框架不仅能够安装到多个版本的 Windows 中,还能够安装到其他智能设备中,这些设备包括智能手机、GPS 导航以及其他家用电器。.NET 框架提供了精简版的应用程序框架,使用 .NET 应用程序框架能够开发容易移植到手机、导航器以及家用电器中的应用程序。

开发人员在使用 Visual Studio 2010 和 .NET 应用程序框架进行应用程序开发时,会发现无论是在原理上还是在控件的使用上,很多都是相通的,这样极大地简化了开发人员的学习过程,无论是 Windows 应用程序、Web 应用程序还是手机应用程序,都能够使用 .NET 框架进行开发。

1.3.3 ASP.NET

ASP.NET 是一个统一的 Web 开发模型,它提供了为建立和部署企业级 Web 应用所必需的服务。同时,ASP.NET 是 .NET 应用程序框架的一部分,是一种可以在高度分布的 Internet 环境中简化应用程序开发的计算环境。当编写 ASP.NET 应用程序的代码时,可以访问 .NET 应用程序框架中的类。可以使用与公共语言运行库(Common Language Runtime,CLR)兼容的任何语言来编写应用程序的代码,这些语言包括 Microsoft Visual Basic、C#、JScript .NET 和 J#。使用这些语言,可以开发利用公共语言运行库、类型安全、继承等方面的优点的 ASP.NET 应用。因此,它有如下特点。

(1) ASP.NET 是同 .NET 应用程序框架集成在一起的,运行在公共语言运行库环境之内。ASP.NET 建立在 .NET 应用程序框架的编程类之上,它提供了一个 Web 应用程序模型,并且包含使生成 ASP Web 应用程序变得简单的控件集和结构。

(2) ASP.NET 是编译执行的,它支持多种编程语言,同时,它也是面向对象的。在 ASP.NET 应用开发中,可以使用与 CLR 兼容的任何语言来编写应用程序的代码,以使用 C# 编写的页面代码为例,它会经过如下两个阶段的编译过程,如图 1.3 所示。

第一阶段,当 ASP.NET 页面第一次被请求时,C# 代码首先被 C# 编译器编译成 .NET 的中间语言 MSIL(Microsoft Intermediate Language)。实际上,不管采用何种 .NET

语言,在此阶段都会被对应的编译器编译成相同的 MSIL 代码,这也是.NET 能够做到语言无关性的关键所在。



图 1.3 页面代码的编译过程

第二阶段,当 ASP.NET 页面实际执行的时候,MSIL 代码会被 JIT(Just-In-Time)编译器编译成机器代码,通常也将此阶段称为即时编译。编译产生的机器码会被存放在缓存中,当页面被再次执行时,代码并没有发生变化,系统将直接从缓存中读取机器码,从而可大大提升执行效率。

(3) ASP.NET 是跨浏览器和跨设备的。如果在开发中完全使用 ASP.NET 自带的 Web 服务器控件,那么这些 Web 服务器控件将会根据客户端的浏览器来自动生成相应的 HTML。这样,不用编写任何额外代码就能够实现跨浏览器支持。

1.4 初识 Microsoft Visual Studio 2010

使用.NET 4.0 框架进行应用程序开发的最好的工具莫过于 Microsoft Visual Studio 2010,Visual Studio 系列产品被认为是世界上最好的开发环境之一。使用 Visual Studio 2010 能够快速构建 ASP.NET 应用程序,并为 ASP.NET 应用程序提供所需要的类库、控件和智能提示等支持。本节将介绍如何安装 Visual Studio 2010 以及 Visual Studio 2010 中窗口的使用 and 操作方法。

1.4.1 安装 Visual Studio 2010

Visual Studio 2010 的安装过程较为简单,按如下步骤操作即可。

(1) 单击 Visual Studio 2010 的光盘或 Visual Studio 2010 试用版(可到微软中国下载 <http://www.microsoft.com/visualstudio/zh-cn/download>)中的 setup.exe 进入安装程序,如图 1.4 所示。

(2) 进入 Visual Studio 2010 安装程序界面后,单击“安装 Visual Studio 2010”链接即可进行 Visual Studio 2010 的安装,如图 1.5 所示。

【提示】在进行安装前,安装程序首先会加载安装组件,这些组件为 Visual Studio 2010 的顺利安装提供了基础保障,在完成组件的加载前不能够进行安装步骤的选择。

(3) 在安装组件加载完毕后,可以单击“下一步”按钮进入安装选项与路径选择窗口,在这里可以进行安装路径的选择并自定义安装组件,如图 1.6 所示。

【提示】在选择路径前,可以选择要安装的功能:“完全”或“自定义”。选择“完全”,将安装 Visual Studio 2010 的所有组件;如果仅需要安装其中的几个组件,可以选择“自定义”进行组件的选择安装。如果仅做 ASP.NET Web 应用程序开发,安装下划线标示的功能即可,如图 1.7 所示。



图 1.4 Visual Studio 2010 安装界面

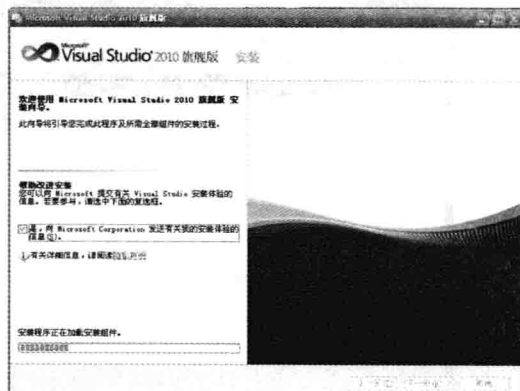


图 1.5 正在加载安装组件

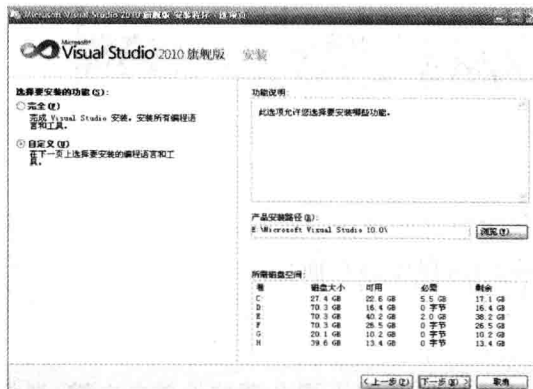


图 1.6 自定义安装功能

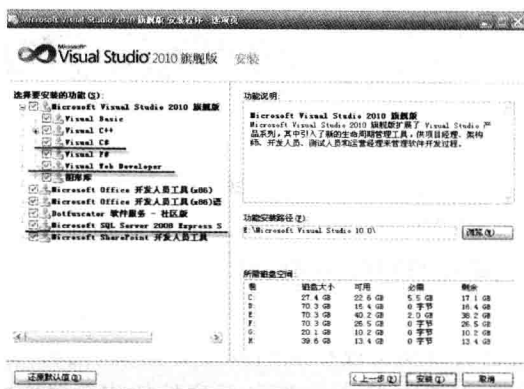


图 1.7 Web 开发必须功能

(4) 路径确定并选择了所需组件后,单击“安装”按钮即可进行 Visual Studio 2010 的安装。

1.4.2 Visual Studio 2010 集成开发环境

1. 主窗口

Visual Studio 2010 极大地提高了开发人员对 .NET 应用程序的开发效率,为了能够快速进入 .NET 应用程序的开发过程,当然需要尽快熟悉 Visual Studio 2010 的开发环境。启动 Visual Studio 2010 后,会呈现如图 1.8 所示的 Visual Studio 2010 主窗口界面。

Visual Studio 2010 主窗口中包括了多个子窗口,可以通过“视图”菜单下相应的菜单项显示或关闭它们。图 1.8 中,左侧为“页面设计与代码编辑”窗口,用于页面的可视化设计及应用程序代码的编写和样式控制;中间为“工具箱”,用于各类服务器控件的存放;右侧为“解决方案资源管理器”窗口,用于呈现解决方案及各种资源文件等,可以将它看做是整个项目的“大管家”。

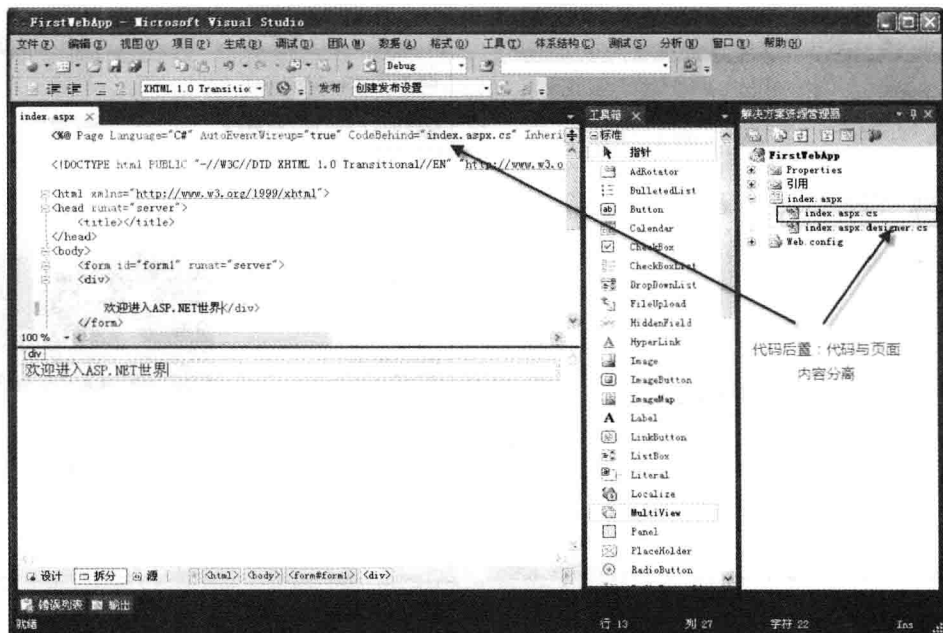


图 1.8 Visual Studio 2010 主窗口界面

2. 页面设计与代码编辑窗口

在 Visual Studio 中,页面设计窗口与代码编辑窗口共享一个窗口,可以在这里设计页面、编写 HTML 代码、编写 C# 代码等。在 Web 应用程序的页面设计中,可以使用“拖曳”的方式将 Web 服务器控件拖曳到页面设计窗口,从而轻松完成页面的布局设计,与此同时它也会自动生成相应的页面 HTML 代码,如图 1.8 所示。

当要修改 Web 服务器控件的相关属性时,只需要在该 Web 服务器控件单击右键,选择“属性”,就可以在属性窗口设置控件的属性了。在这里还可以为控件添加相应的事件。当然,也可以通过鼠标双击控件的方法来为控件添加事件,如图 1.9 所示。

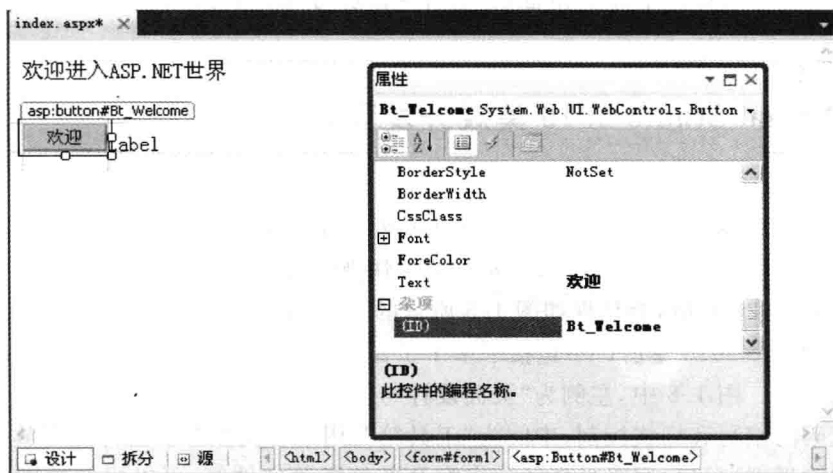


图 1.9 选择“设计”选项及控件的属性设置窗口

图 1.9 中,页面设计窗口下面有 3 个按钮:设计、拆分和源。它们是 Visual Studio 提供的 3 种 Web 页面的设计模式,分别适合于不同使用习惯的设计人员与不同的阶段需求。

(1) 设计模式:它提供可视化的页面设计,控件元素拖曳上去后就能够马上看到设计效果,很适合那些对 HTML 代码书写不熟练的设计人员或在界面设计阶段采用。

(2) 源模式:它是提供纯 HTML 代码方式的设计模式,适用于那些对 HTML 代码比较熟悉的设计人员或在界面的修改、完善阶段采用。

(3) 拆分模式:如图 1.8 左侧窗口所示,它合并了上面两种设计模式,让我们既能够看到页面的设计效果,又能够了解其对应的 HTML 代码,可在完成页面布局的同时学习对应的 HTML 代码,非常适合于初学者采用。

3. 工具箱

工具箱是 Visual Studio 2010 中的基本窗口,可以使用工具箱中的控件进行应用程序的快速开发。在“视图”菜单下单击“工具箱”菜单项,可以在主窗口中显示如图 1.10 所示的“工具箱”窗口,其中包含了如图 1.11 所示的默认选项卡类别。

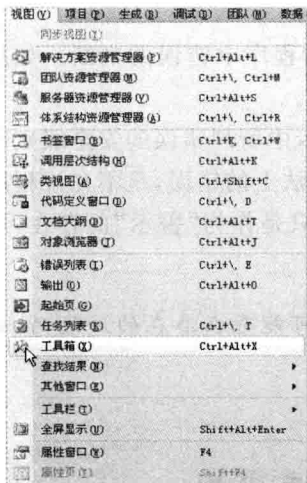


图 1.10 工具箱

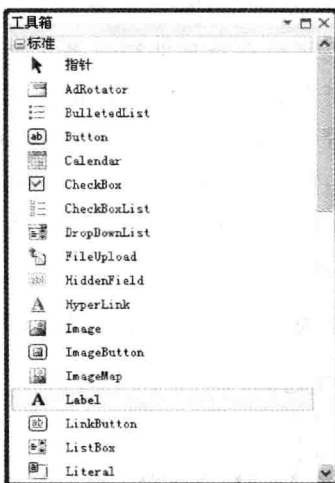


图 1.11 默认选项卡

控件工具箱中包含了 Web 应用程序开发时常用的控件,也被看做 Visual Studio 的一大特色。利用工具箱中的控件,我们无须编写任何代码,使用鼠标“拖曳”的操作方式即可完成页面的界面设计,而且这些控件都可以做到跨浏览器和跨设备运行。

在对工具箱控件有所熟悉后,还可以按照我们的使用习惯自定义工具箱的选项卡名(如标准、数据等)以及选项卡中的控件项。可以右击选项卡标题来选择“重命名选项卡”、“添加选项卡”或者“删除选项卡”,在工具箱的空白处右击并选“选择项”就可以添加一个或者多个项。当然,也可以把一个项从一个选项卡拖放到另一个选项卡内。

4. 解决方案管理器

为了能够方便用户进行 Web 应用程序的开发,默认情况下,在主窗口的右侧会呈现一个解决方案管理器,如图 1.8 右侧窗口所示。在这里,不仅可以查看整个项目的项目文件,还可以管理项目解决方案,并在项目解决方案下根据设计需要随时添加、修改、删除子项目