

高等院校信息技术应用型特色教材



ASP.NET Web应用程序设计及开发(C#版)

邵鹏鸣 编著



清华大学出版社

高等院校信息技术应用型特色教材

ASP.NET Web应用程序设计及开发(C#版)

邵鹏鸣 编著

清华大学出版社

北 京

内 容 简 介

本书通过一个能够应用于实际的电子商务站点案例为主线,深入浅出、全面而详尽地讲解了.NET Web 应用程序设计的基本方法、技巧及注意事项,目的是培养学生编写实际 Web 应用程序的能力,其基本内容包括 Web 窗体、ADO.NET 技术、Web 窗体数据绑定、服务器控件、Web 服务、服务器控件窗体验证、身份验证、Web 用户控件、Web 窗体状态管理、样式和自定义控件。

本书通过多个详尽的案例分折,使学生能够尽快地掌握 ASP.NET 的关键技术。此外,书中提供了大量与开发相关的技术要点提示,同时配合实用、有效的 Web 应用程序,使学生能够迅速掌握并巩固所学到的知识。本书汲取了很多来自实际编程中的体验,这将为培养良好的编程习惯打下一个坚实的基础。

本书可作为高职高专计算机专业和高等技术型院校 Web 编程与开发课程的教材,也可供程序设计开发人员参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13501256678 13801310933

图书在版编目(CIP)数据

ASP.NET Web 应用程序设计及开发(C#版)/邵鹏鸣编著. —北京:清华大学出版社, 2007. 10

高等院校信息技术应用型特色教材

ISBN 978-7-302-15608-6

I. A… II. 邵… III. ①主页制作—程序设计—高等学校—教材②C 语言—程序设计—高等学校—教材 IV. TP393.092 TP312

中国版本图书馆 CIP 数据核字(2007)第 098374 号

责任编辑:孟毅新

责任校对:李 梅

责任印制:何 芊

出版发行:清华大学出版社 地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn> 邮 编:100084

c-service@tup.tsinghua.edu.cn

社总机:010-62770175 邮购热线:010-62786544

投稿咨询:010-62772015 客户服务:010-62776969

印 刷 者:北京密云胶印厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260 印 张:24.75 字 数:566 千字

版 次:2007 年 10 月第 1 版 印 次:2007 年 10 月第 1 次印刷

印 数:1~4000

定 价:32.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:024140-01

前言

ASP.NET Web 应用程序设计及开发(C#版)

随着 21 世纪的到来,计算机技术的发展更加迅猛,在各行各业的应用也更加广泛。面对日新月异的新技术、新方法,我们必须对现有计算机课程的设置和教学内容进行调整,以适应技术进步与市场变化的需要。

今天,应用程序已由驻留在用户硬盘上的独立可执行文件发展为由 Web 服务器在 Internet 上传送的分布式应用程序,相应地,任何一种开发平台及程序设计语言都必须适应这种变化。微软的 .NET 是一种开发平台,ASP.NET 是 .NET 的组件之一,它为创建 Web 应用程序提供了一种新的编程模型和结构。Visual Basic .NET 是在 .NET 平台上编程的一种高级语言,它为开发人员提供了生成企业级 Web 应用程序所需的服务。

使用 .NET 创建的 Web 应用程序都可以使用整个 .NET Framework。开发人员可以方便地获得这些技术的优点,其中包括托管的公共语言运行库环境、类型安全、继承,.NET Framework 类库等。

ASP .NET 可以无缝地与所见即所得的 HTML 编辑器和其他编程工具(包括 Microsoft Visual Studio .NET)一起工作,这使 Web 开发更加方便,并且包括开发人员可以用来将服务器控件拖放到 Web 页的 GUI 控件和完全集成的调试支持。当创建 Web 应用程序时,开发人员可以使用 Web 窗体或 Web 服务,并可以像创建 Windows 应用程序那样使用事件驱动的编程方法。

使用 .NET 可创建出更安全、可伸缩和稳定的企业级 Web 应用程序。

微软以 .NET 平台和 C#、Visual Basic .NET 编程语言及 ASP .NET 的形式与 J2EE 和 Java 进行全面的竞争。这是微软在编程语言和编程环境这块竞技场上所做出的最出色的成果。当然,他们有相当有利的条件:他们可以看到 Java 在什么方面做得更好,在什么方面做得还不够好,然后,基于此去构建,并要具备 Java 不具备的优点。

目前,与 J2EE 相比,对 .NET 平台人们所关心的最重要的问题是:微软是否会将它完全移植到其他平台上。微软宣称这没有问题,而且 Mono 项目(www.go-mono.com)已经有了一个在 Linux 上运行的 .NET 的部分实现。

学习一种先进的语言和一种先进的编程方法将激起学生更大的兴趣。这些知识在他们离开学校,进入一个由信息占据重要地位的社会时可以立即发挥作用。正是这一点激发了他们对知识的学习热情。运用 ASP .NET 能够出色地完成任务,所以他们更愿意投入更多的精力和时间。

很多院校都有软件专业的学生,有的学校的学生在几年的学习时间里,不仅学了 C、

C++，而且也学了 Java、Visual Basic 或 Visual Basic.NET 甚至 C#，同时还学习了 Web 应用程序设计及其他有关软件开发的课程。但据作者的调查，有些院校软件专业的学生在毕业找工作时，很难找到一份满意的工作，他们的工作与软件无关，甚至与计算机无关。不是企业不需要软件专业的人才，而是没有所需要的人才。我认识一位学生，学习成绩不错，学习也很勤奋，在毕业前夕还通过了全国计算机技术与软件专业技术资格(水平)考试，获得软件设计师资格。但当我交给他一份简单的实际任务时，他根本无从下手。这些都说明，我们的教学体系、教学内容、教学模式和方法都存在很多需要改进的地方。

本人不赞成用“玩具例程”说明概念、原理的教学方法。这些例子完全脱离实际，看起来很幼稚，尽管它能够说明一个概念，学生也容易看懂，上课也容易听懂。但是检查学习的好坏的标准，不是“知道还是不知道，听懂还是没有听懂，看懂还是没有看懂”，而是“会不会应用”。真实例程同样能够说明概念、原理和知识点，只不过是面对复杂的技术实践体系，需要花大量的时间、精力去设计这样的例程。只要设计得好，真实例程同样能够说明概念、原理和知识点，学生同样容易看懂、听懂，并且懂得在实际应用中如何运用这些概念、原理和知识点，起到事半功倍的效果。这就是本人编写本书的目标之一。

我的目标很明确，写一本实用、有一定深度且易读的 ASP.NET 程序设计教材，无论读者有无编程经验，要做到开卷有益，重要的是能够激发他们的学习兴趣，能够将他们所学的知识应用于实际。

本书将基本技能培养和主流技术相结合，以市场对人才的需求为依据，从应用与工程实践的角度出发进行组织，其主要特点如下。

1. 新体系、新内容、新手段、新思路。无论是内容体系、编写的思路、教学的模式及理念都具有一定的新意。

2. 先进性及实用性。本书的内容反映最新的、实用的程序设计方法及技术，符合新世纪教学发展的规律。书中讲授的程序设计方法与现代编程方法步调一致，具有很强的实用性。

3. 面向实际的应用程序驱动的教学方式。本书不是采用传统的“提出概念—解释概念—举例说明”的方法，而是以实践为主线，以应用为目标，通过完成实际的应用程序的方式学习程序设计知识。本书采用“提出任务—介绍完成任务的方法及步骤—归纳出一般的规律、概念及知识点”的模式。每一个新概念、知识点的提出都伴随着一个完整的、可实际运行的实用程序及其输入、输出。学生通过完成实例，掌握概念和知识点。

4. 使读者从一开始就编写有用的程序，这有助于激发读者的学习兴趣及动力。书中的实例不是只有几行代码的小程序，这些实例来源于实际应用程序或其中的一部分，它们是与我们的生活相关或大家感兴趣的内容。本书包含大量这样的实例，这些实例及其代码分析与讨论是本书的精髓。读者可先按照所描述的方法和步骤编写代码并运行自己所创建的应用程序，“试探”该应用程序，了解其工作过程，然后通过其代码分析与讨论掌握其包含的概念、原理和方法。

5. 逐步展现主题鲜明的各章。本书将集中重要的主题并进行充分论述，而不是肤浅地涉及许多问题，罗列许多概念、术语、语法。本书的内容及体系结构使学生感到它既具有易读性，又增长知识，具有很强的实用性。

6. 面向实际的技术。学习程序设计的目的是开发程序。本书不只是讲授语言要素、语法,而是教会读者如何用 ASP.NET、C# 开发 Web 应用程序,书中的每一个面向实际的应用程序实例都说明了开发过程及包含的知识点。通过学习如何开发程序来掌握语言要素、语法。本书通过一个实际的电子商务网站应用程序实例,讲授了三层架构的应用程序设计方法和开发过程。三层应用程序设计是企业应用程序常用的程序设计方法,作者通过这种方式,使读者能够熟练应用这种企业常用的程序设计方法和开发过程。

7. 面向对象的程序设计。本书采用了面向对象编程的思想。我们的目标之一是使学生习惯于现实世界的程序设计,仅仅知道面向对象的概念是不够的,学生们必须能够运用这些知识开发现实世界的程序。

8. 循序渐进,由浅入深,综合应用。本书在设计时贯穿了一个思想:面向实际,同时使读者容易理解、明白所包含的所有细节。因此本书在安排时,循序渐进,由浅入深,后序章节在讲授新的知识点的同时综合应用了前面章节所讲的知识。这样做既温习了前面所讲的知识,同时也讲解了如何综合应用前面所学的知识解决问题。在书中很多实际应用程序实例中,不是只讲授一个新的知识点,而是在讲授新知识点的同时,应用了前面所讲过的知识。

9. 数据库。如今数据库应用程序对所有企业来说是至关重要的,因此本书在第二章用一章的篇幅,通过一个具体的实际应用程序实例讨论使用 ADO.NET 与 SQL 访问数据库的编程技术,为后续学习打下一个基础。后续章节在构建电子商务站点时,都应用了 ADO.NET 与 SQL 访问数据库的编程技术,并应用了三层架构的应用程序设计方法。

我们希望读者遵循书中介绍的方法和步骤实际建立实例程序,然后将实例程序进行修改或扩展,并通过对实例的代码进行分析与讨论,掌握实例背后包含的概念、原理、知识点和方法等,我们认为这是学习程序设计最稳妥、最有效、最快捷的途径。

通过本书的学习,读者应能达到如下目标:

- 面向对象的 Web 应用程序设计;
- 三层架构的应用程序设计方法和开发过程;
- ADO.NET 及 Web 数据库应用程序设计;
- 开发一个完整实用的电子商务站点。

由于水平的关系,书中的错误在所难免。如果发现不当之处,欢迎提出宝贵意见 (pmsbao@163.com),作者不胜感激。

邵鹏鸣

2007 年 9 月

目 录

ASP.NET Web 应用程序设计及开发(C#版)

第 1 章 认识 ASP.NET	1
1.1 什么是 .NET	1
1.1.1 从用户代码到机器代码	3
1.1.2 两种中间语言介绍	3
1.2 什么是 ASP.NET	4
1.3 什么是 Web 窗体	4
1.4 创建第一个 Web 应用程序	5
1.4.1 创建项目和窗体	5
1.4.2 检查 Web 窗体结构	6
1.4.3 添加控件和文本	7
1.4.4 创建事件处理程序	11
1.4.5 生成并运行 Web 窗体页	11
1.4.6 ASP.NET 网页的回发和往返行程	19
第 2 章 HTML 服务器控件	20
2.1 HTML 服务器控件共有的属性	20
2.2 按钮、标签和文本框	22
2.2.1 按钮	22
2.2.2 标签	22
2.2.3 文本框和密码框	22
2.3 单选按钮、复选框和面板控件	24
2.3.1 单选按钮	24
2.3.2 复选框	25
2.3.3 面板控件	25
2.4 下拉列表框、列表框和图像控件	28
2.4.1 下拉列表框和列表框	28
2.4.2 图像控件	29
第 3 章 使用 ADO.NET 访问数据库	32
3.1 Connection 和 Command 对象	32

3.1.1	Connection 对象	33
3.1.2	RadioButton 控件的常用属性和事件	38
3.2	Command 对象	38
3.2.1	使用 Command 对象直接执行数据库操作	38
3.2.2	TextBox 服务器控件	42
3.3	使用 ADO.NET SqlDataReader	42
3.3.1	使用 SqlDataReader 装载列表框	43
3.3.2	使用数据表的记录创建 ListItem 对象	46
3.3.3	使用参数化查询	49
3.3.4	装载下拉列表框	51
3.3.5	在下拉列表框中查找值	53
3.3.6	列表控件(列表框、组合框)的常用属性和事件	55
3.3.7	修改数据	56
3.4	使用 DataAdapter、DataTable 和 DataSets	58
3.4.1	DataTable 对象	58
3.4.2	数据集和数据适配器	61
3.5	修改数据	68
3.5.1	使用数据集向数据库表中添加一行	68
3.5.2	修改数据集中表中的行	71
3.5.3	在数据集中删除行	74
第 4 章	Web 窗体数据绑定	77
4.1	数据绑定概述和语法	77
4.1.1	Web 窗体页的多记录数据绑定	78
4.1.2	单值数据绑定	79
4.1.3	在设计时绑定与在运行时绑定	79
4.1.4	控件属性的数据绑定表达式	79
4.2	控件绑定简单属性	80
4.2.1	在设计时控件绑定简单属性	80
4.2.2	在运行时控件绑定简单属性	85
4.3	DataBinder、Eval 方法	87
4.4	实现导航按钮	93
第 5 章	列表控件	96
5.1	DataGrid 控件	97
5.1.1	使用 DataGrid 显示表格数据	97
5.1.2	使用 DataGrid 编辑数据	102
5.1.3	更新数据集和数据库	105

5.1.4	类型化数据集与非类型化数据集	107
5.1.5	从数据表中删除记录	108
5.1.6	在 DataGrid 中分页	109
5.1.7	DataGrid 控件的数据绑定、常用属性及事件	111
5.2	DataList Web 服务器控件	113
5.2.1	使用 DataList 中的项目模板和选择项目模板	114
5.2.2	通过类编号查询产品	120
5.2.3	从模板处理回发	124
5.2.4	Image 和 HyperLink Web 服务器控件	134
5.2.5	DataList 控件	135
5.2.6	会话状态	139
5.3	按钮服务器控件	142
5.3.1	按钮控件的常用属性	142
5.3.2	按钮控件的常用事件	143
5.3.3	容器控件中的按钮	144
5.4	Repeater Web 服务器控件	144
5.4.1	在 Web 窗体页中创建只读数据访问	145
5.4.2	创建类别菜单	149
第 6 章	创建多层的分布式应用程序	153
6.1	设计方案	153
6.2	数据库层	155
6.2.1	创建购物车表	155
6.2.2	创建执行基本的购物车操作的存储过程	156
6.3	业务层	159
6.3.1	创建业务层	159
6.3.2	Cookie	168
6.3.3	HttpContext 类	172
6.4	表示层	173
6.4.1	创建 Web 窗体	173
6.4.2	自定义 DataGrid 控件中的列	175
6.4.3	导入业务对象	182
6.4.4	编写代码	182
6.4.5	修改 Web 窗体页 AddToCar.aspx	186
6.4.6	测试应用程序	187
第 7 章	Web 服务	189
7.1	了解 Web 服务	189

7.1.1	概述	189
7.1.2	什么是 Web 服务	190
7.2	创建 Web 服务	190
7.3	访问 Web 服务	195
7.4	简单对象访问协议(SOAP)和 Web 服务	199
7.5	搜索类目	200
7.5.1	创建 CatalogService Web 服务	201
7.5.2	访问 CatalogService Web 服务	205
第 8 章	服务器控件窗体验证	210
8.1	RequiredFieldValidator 与 ValidationSummary 控件	211
8.1.1	客户登录	211
8.1.2	RequiredFieldValidator 控件及其常用属性	226
8.1.3	ValidationSummary 控件及其常用属性	228
8.2	CompareValidator 与 RegularExpressionValidator 控件	229
8.2.1	客户注册	229
8.2.2	CompareValidator 服务器控件及其常用属性	241
8.2.3	RegularExpressionValidator 服务器控件及其常用属性	242
8.3	客户结账	242
8.3.1	创建数据层	244
8.3.2	创建业务层	245
8.3.3	创建表示层	250
8.3.4	设计订单 Web 窗体页	264
8.4	视图状态	269
8.4.1	视图状态的作用	269
8.4.2	以视图状态存储信息	269
8.4.3	从视图状态检索信息	269
8.4.4	EnableViewState 属性	269
8.5	CustomValidator 服务器控件	270
8.5.1	概述	270
8.5.2	结账页面的自定义验证	270
8.6	客户端验证	271
第 9 章	基于窗体的身份验证	274
9.1	登录、账户管理及结账	274
9.2	启用 Web 应用程序以使用 Forms 身份验证	282
9.3	授权用户和角色	284
9.4	Forms 身份验证凭据	286

9.5 Forms 身份验证实用工具	287
9.5.1 FormsAuthentication. Authenticate 方法	287
9.5.2 FormsAuthentication. HashPasswordForStoringInConfigFile 方法	287
9.5.3 FormsAuthentication. SetAuthCookie 方法	289
9.5.4 FormsAuthentication. GetAuthCookie 方法	290
9.5.5 FormsAuthentication. RedirectFromLoginPage 方法	291
9.5.6 FormsAuthentication. GetRedirectUrl 方法	292
9.5.7 FormsAuthentication. SignOut 方法	293
9.6 在 Web.config 中存储数据	293
第 10 章 Web 用户控件	295
10.1 创建 Web 用户控件	295
10.1.1 编写业务层代码	297
10.1.2 用户控件界面设计	297
10.1.3 用户控件的初始化及事件处理	300
10.2 使用用户控件	305
10.3 公开用户控件属性	310
10.4 操作用户控件属性	311
10.5 使用 HtmlInputFile 服务器控件	311
10.5.1 产品管理	312
10.5.2 添加产品	325
10.6 以编程方式向 Web 窗体页添加用户控件	328
第 11 章 Web 窗体状态管理	333
11.1 应用程序状态	334
11.1.1 聊天室	334
11.1.2 关于多线程的说明	338
11.2 应用程序和会话状态事件	338
11.3 缓存应用程序数据	343
11.4 有关状态的比较	345
11.4.1 视图状态	345
11.4.2 Cookie	346
11.4.3 查询字符串	346
11.4.4 会话状态	346
11.4.5 应用程序状态	347

第 12 章 对控件应用样式	348
12.1 使用 CSS 样式	348
12.1.1 级联式样式表(CSS)介绍	348
12.1.2 在 STYLE 块内定义 CSS 样式规则	348
12.1.3 CSS 样式规则的优先级	349
12.1.4 将 HTML 元素分配给 CSS 样式 CLASS	349
12.2 对 HTML 控件应用样式	352
12.3 应用外部 CSS 样式表	354
第 13 章 自定义控件	361
13.1 概述	361
13.2 创建第一个 Web 自定义控件	362
13.2.1 创建 Web 应用程序项目	363
13.2.2 创建自定义控件	363
13.2.3 使用自定义控件	364
13.2.4 创建自定义设计器	365
13.2.5 个性化控件	368
13.3 维护控件中的状态	369
13.4 开发复合控件	372
13.4.1 概述	372
13.4.2 创建 Login 控件	372
13.4.3 在页上使用 Login 控件	374
13.5 在复合控件中处理事件	376
13.5.1 创建 Login 控件	376
13.5.2 在页上使用 Login 控件	378
13.6 从复合控件触发自定义事件	379
13.6.1 修改 Login 控件	379
13.6.2 在页上使用 Login 控件	380
参考文献	382

认识 ASP.NET

本章主要内容

1. 什么是 .NET。
2. 什么是 ASP.NET。
3. 什么是 Web 窗体。
4. 创建 Web 窗体页。
5. 将控件和静态 HTML 文本添加到页。
6. 将 HTML 元素转换为服务器控件。
7. 认识 HTML 服务器控件。
8. 认识 Web 窗体服务器控件。
9. 事件及事件处理程序。
10. 为控件创建事件处理程序。
11. 生成并运行 Web 窗体页。
12. 熟悉 Web 窗体结构。
13. 熟悉集成开发环境。
14. Web 窗体控件事件模型。
15. Web 窗体代码模型。

1.1 什么是 .NET

.NET 是微软公司的新战略,它包含微软公司对未来的核心战略、规划和洞察力。所有微软公司的产品都将围绕这个战略开发。此战略的核心就是 .NET Framework,该框架提供了全面支持 .NET 的核心技术。.NET Framework 是一种新的计算平台,它简化了在高度分布式 Internet 环境中的应用程序开发。.NET Framework 具有两个主要组件:公共语言运行库和 .NET Framework 类库。

.NET 用来解决编程人员面临的许多问题,例如:

- ① 负责处理在创建大型、可靠的应用程序时的大量艰辛工作。
- ② 允许程序员统一两种架构——在本地机器上运行的应用程序和通过 Web 访问的

应用程序。

③ 减少了与编程框架相关的传统开销——不再需要高性能编程语言来编写复杂的代码以获取高性能的 .NET 程序。

④ 允许不同语言的程序员在一个应用程序中协同工作。

⑤ 开始兼容各种最终用户工具。包括桌面、PDA 和手机。使开发人员能够创建出摆脱设备硬件束缚的,能够在各种操作系统上运行的应用程序。能够轻松实现互联网的连接。

总之,.NET 提供了一种更简单、更快捷、更廉价的方式,来获得高效的程序。

在某些方法上,.NET 很像 Java。实际上,Java 的口号是“一旦编写出来,就能在任何地方运行”,很符合 .NET 的原则。不过 .NET 并不是 Java 的克隆,微软的方法与之并不相同。

Java 是“一种语言,多个平台”,而 .NET 是“多种语言,一个平台……(就现在而言)”,微软想扫除进入 .NET 的障碍,为此就要让使用任何语言的人都可对它进行访问。.NET 有两种语言,即 C# 和 Visual Basic.NET,而 Visual Studio.NET 提供了这两种语言。

微软的策略更像是一场军事战役。它运用对 Windows 平台的理解,首先攻占 Windows 平台。当它从 Windows 平台中脱颖而出后,就可以“侵袭”别的平台(例如 Linux),证明 .NET 应用程序可以从一个平台移植到另一个平台上。“侵袭并征服了 Linux”后,它又可转向另外的平台,例如 Solaris。

简而言之,.NET 将操作系统平台割裂开来。无论是哪种平台,如 Windows、Linux 等,都可以分成两个层次:程序设计层和执行层。.NET 开发人员是为程序设计层而不是执行层编写软件,将来不管占统治地位的是 Windows 平台还是 Linux 平台,或者是其他出乎意料的系统,就无关紧要了。

可以将 .NET Framework 的讨论分解成以下几部分。

① MS 中间语言(MS Intermediate Language):在执行用户编写的所有程序代码前,应将其编译成更抽象、精简的形式。程序员可以使用任何 .NET 语言编写代码,包括 Visual Basic、C#、JScript 和其他大约 20 种语言。这些代码都会编译为 MSIL 这种 .NET 的通用语言。.NET 在这个层次的操作无需用户干涉,因此本书不讨论它。

② 公共语言运行时(Common Language Runtime,CLR):这是一个复杂的系统,用于在计算机上执行 MSIL 代码。公共语言运行时负责执行与 Windows 和 IIS 交互时所涉及的全部实质性任务。这个层次也是在后台进行的,本书也不涉及。

③ .NET Framework 类库(.NET Framework Class Libraries):实现大量重要功能的代码库。用户可以非常方便地在应用程序中调用库函数,使复杂任务的程序代码更为简洁。本书将介绍这些功能。

④ .NET 语言(.NET Language):.NET 语言是符合一些特殊结构要求(由公共语言规范定义)的编程语言,能够编译成 MSIL。可以用任一种语言进行开发,例如 C# 或 Visual Basic.NET,这没有任何限制,也可以用多种语言来开发程序。本书将用几章的篇幅讨论 Visual Basic 在 ASP.NET 中的应用,并在所有的代码示例中使用 Visual Basic。

⑤ ASP.NET:这个代码模块扩展了 Internet Information Server(IIS),使之能为

Web 页面执行 .NET Framework。本书将涉及几乎所有的 ASP.NET 功能。

⑥ Web 服务: 虽然 Web 服务不是严格意义上的 .NET 的一部分, 但 .NET 明确地支持它。它们是可以通过 Web 访问的组件, 可以是任何主题, 例如新闻标题、天气预报、股票走势、病毒预防和操作系统更新等。第 7 章将详细讨论 Web 服务。

在详细介绍上述内容之前, 首先介绍一些基本的代码概念与术语。

1.1.1 从用户代码到机器代码

计算机识别以二进制形式表示的任何事物(二进制位是表示指令和数据的 1 与 0 的序列)。因此, 人们热衷于用“数字”一词来描述任何事情, 即使它们与计算机的关系很小。通常把这种二进制位指令称为机器码。很明显, 对大多数人来说, 记住用于打印“Good Morning”时的 0 和 1 的代码序列是不可能的, 更不用说要记住定义复杂 Web 应用程序的程序代码了。因此, 人们发明了编程语言, 通过类似英语的单词来编写代码。

一旦用高级语言编写了程序代码, 就需要将其转换为机器码, 这一转换过程称为编译, 编译器软件就可以将人类可读指令编译成机器可读指令。编译过程包括将本机环境信息写入经编译的程序代码这一过程, 因此编译后的程序代码可以最高效地利用计算机的所有可用资源。

多年来, 有以下两种编译类型, 它们的编译过程完全不同。

① 预编译型代码——在编码完成后并在执行前进行编译的代码。对于这种代码, 由于编译器会花时间分析全部的代码和要运行它的计算机, 因此代码能以很快的速度执行。不过, 由于预编译的代码是针对某一计算机的, 因此, 除非仍在该计算机上使用编译过的代码, 否则需要将另一台计算机设成与第一台计算机相同的系统与资源。

② 解释型代码——边执行(在用户请求页面时)边编译的代码。这种代码执行较慢, 因为需要为每个请求编译代码, 且系统没有机会全面优化所编写的代码。其优点是解释过程可以调整, 以适应运行代码的计算机。

所以, 开发人员在选择语言方面要做出取舍。既可以选择较慢的解释性代码, 以适应运行代码的计算机, 也可以选择较快的预编译代码, 但不能充分利用计算机的优势。

1.1.2 两种中间语言介绍

.NET 在编译时采用两步来解决这个问题。编写在 .NET Framework 上运行的程序时(通常用 Visual Basic .NET 或 C# 编写), 需要在使用这些程序之前编译这些可读代码。.NET 编译器的设计方式意味着它并没有把我们带入会引起可移植性问题的二进制码。事实上, .NET 编译器将程序代码编译成称为 MS 中间语言(MSIL)的特殊格式。由于 MSIL 结构不需要像源代码那样易读, 因此编译过程包括了一些优化操作。但是, 并没有针对某台计算机进行优化。因此, MSIL 具有一般的优化性能, 并可以移植到任何 .NET 服务器上。

当用户执行 MSIL 代码时(即请求 ASP.NET 页面), 将其传给 CLR, CLR 是 .NET Framework 的另一个核心。CLR 使用 JIT(Just-In-Time)编译器将代码编译成真正的机器码, 并对程序进行最后的且与计算机相匹配的优化, 以使程序能在其所在的计算机上以尽可能快的速度运行。

MSIL 和 CLR 组合使用,具有两种编译代码的优点,即预编译码的结构优化和解释码的可移植性。

更为重要的是,MSIL 本身与计算机无关。因此,可以在装有 CLR 的任一计算机上运行。实际上,一旦编写出 .NET 程序代码并将其编译,就可以将它复制到装有 CLR 的计算机上,并在该计算机上执行。虽然目前的 CLR 只有与 Windows 兼容的版本,但现在已着手建立基于其他操作系统的版本。在 Web 上搜索与 Mono Project 的信息,就可以找到更多的 CLR 版本。

MSIL 也可以由任一遵循 CLS 的可读语言生成。Visual Basic .NET、C# 和 JScript.NET 是“与 .NET 兼容的”三大语言。另外 MSIL 编译器还支持其他 20 多种语言。因此,可以在应用程序内部交替地使用这些兼容语言。一旦将一套文件编译成 MSIL,它们都将统一为一种语言。这种灵活性允许不同的小组在同一个 Web 站点上用不同的语言协同工作。

1.2 什么是 ASP.NET

ASP.NET 是一个基于 .NET 的统一的 Web 开发平台,该 Web 开发平台使得 Web 开发人员可以使用任何 .NET 编程语言(包括 Visual Basic .NET、C# 和 C++ 托管扩展)开发 Web 应用程序。ASP.NET 还是 .NET Framework 的一部分,所以它提供对该框架所有功能的访问。

1.3 什么是 Web 窗体

Web 窗体是一项 ASP.NET 功能,我们可以使用它为 Web 应用程序创建用户界面。Web 窗体页提供了一种强大而直接的编程模型,该模型使用快速应用程序开发(RAD)技术来生成基于 Web 的复杂用户界面。

在 Web 窗体页中,用户界面编程分为两个不同的部分:可视组件和逻辑。如果用户以前使用过类似于 Visual Basic 和 Visual C++ 的工具,将认同在窗体的可视部分和窗体后与之交互的代码之间存在这样一种划分。

视觉元素被称作 Web 窗体“页”(page)。Web 窗体页由一个包含静态 HTML 或 ASP.NET 服务器控件的文件组成。文件的扩展名为 .aspx。

Web 窗体页用作要显示的静态文本和控件的容器。利用 Visual Studio Web 窗体设计器和 ASP.NET 服务器控件,我们可以按照在任何 Visual Studio 应用程序中的方式来设计窗体。

Web 窗体页的逻辑由代码组成,这些代码由用户创建以与窗体进行交互。编程逻辑位于与用户界面文件不同的文件中。该文件称作“代码隐藏”文件,扩展名为“.aspx.cs”。在代码隐藏文件中编写的逻辑可以使用 Visual Basic 或 Visual C# 来编写。

1.4 创建第一个 Web 应用程序

实例 1-1 Hello, World

主要知识点：创建一个新的 Web 窗体页，将控件和静态 HTML 文本添加到页，将 HTML 元素转换为服务器控件；认识 HTML 服务器控件，认识 Web 窗体服务器控件，事件及事件处理程序，为控件创建事件处理程序，生成并运行 Web 窗体页，熟悉 Web 窗体结构，熟悉集成开发环境，Web 窗体控件事件模型，Web 窗体代码模型。

创建图 1.1 所示 Web 应用程序，当单击“显示”按钮时，在文本框中显示“Hello, World”，单击“清除”按钮时，将文本框中的内容清空。



图 1.1 第一个 Web 应用程序

1.4.1 创建项目和窗体

- 1) 在“文件”菜单上指向“新建”，然后选择“项目”命令。
- 2) 在“新建项目”对话框中(如图 1.2 所示)，执行以下操作：
 - (1) 在“项目类型”窗格中选择“Visual C# 项目”。
 - (2) 在“模板”窗格中选择“ASP.NET Web 应用程序”。
 - (3) 在“位置”文本框中，为应用程序输入完整的 URL(包含 http://、服务器名称和项目名称)。Web 服务器上必须安装 IIS 5(或更高版本)和 .NET Framework。如果计算机上已安装 IIS，可以为服务器指定 http://localhost。

注意：如果已经打开了一个解决方案，可选择“关闭解决方案”，从而使新的 Web 窗体项目成为它自己解决方案的一部分。

- 3) 单击“确定”按钮后，将在指定的 Web 服务器的根处创建新的 Web 窗体项目。

此外，名为 WebForm1.aspx 的新 Web 窗体页将显示在“设计”视图中 Web 窗体设计器上(如图 1.3 所示)。