

Web程序设计

——ASP.NET网站开发

- ◆ ASP.NET基础及开发环境构建
- ◆ C#语言入门基础
- ◆ ASP.NET常用控件功能及其应用
- ◆ ASP.NET常用对象功能及其应用
- ◆ 访问数据库
- ◆ 网站设计技术
- ◆ LINQ技术
- ◆ ASP.NET AJAX技术
- ◆ 文件操作
- ◆ ASP.NET动态网站开发



陈作聪 王永皎 程凤娟 编著



清华大学出版社

高等学校计算机应用规划教材

Web 程序设计

——ASP.NET 网站开发

陈作聪 王永皎 程凤娟 编著

清华大学出版社

北 京

内 容 简 介

本书由浅入深、循序渐进地介绍了如何使用 ASP.NET 4.0 和配套的 Visual Studio 2010 开发环境进行 Web 网站开发所要掌握的各种技术、操作方法和使用技巧。全书共分 13 章, 分别介绍了 ASP.NET 4.0 基础知识、C# 入门知识、Web 服务器控件、主题、母版页、网站导航、数据绑定、数据库的访问、LINQ 查询技术、Web 服务和 ASP.NET AJAX 等 Web 网站开发必须了解的各种知识。

为了方便读者自学和教师教学, 本书配有实验指导书《Web 程序设计——ASP.NET 上机实验指导》(ISBN: 978-7-302-27923-5)。

本书内容丰富, 结构清晰, 语言简练, 图文并茂, 具有很强的实用性和操作性, 是一本适合于高等院校 Web 程序设计课程的优秀教材, 也是广大初、中级软件开发爱好者的自学参考书。

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 13701121933

图书在版编目(CIP)数据

Web 程序设计——ASP.NET 网站开发/陈作聪, 王永皎, 程凤娟 编著. —北京: 清华大学出版社, 2012.3
(高等学校计算机应用规划教材)

ISBN 978-7-302-27534-3

I. W… II. ①陈… ②王… ③程… III. 网页制作工具—程序设计—高等学校—教材
IV. TP393.092

中国版本图书馆 CIP 数据核字(2011)第 268658 号

责任编辑: 王 定 胡花蕾

装帧设计: 牛艳敏

责任校对: 蔡 娟

责任印制: 何 芊

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 北京密云胶印厂

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 22.25

字 数: 514 千字

版 次: 2012 年 3 月第 1 版

印 次: 2012 年 3 月第 1 次印刷

印 数: 1~5000

定 价: 35.00 元

前言

随着网络技术的不断发展,如何能够快速而高效地开发出 Web 网站已经成为编程人员所共同关注的问题。为了适应广大编程者对网站开发的要求,Microsoft 公司在 2010 年推出了最新的 ASP.NET 版本——ASP.NET 4.0。它是新一代 Web 应用程序开发平台和框架,实现了企业级 Web 应用程序的快速开发,通过提供最简单、最具可扩展性的方式,来开发、部署和运行任何浏览器或客户端设备作为目标的 Web 应用程序。同时,ASP.NET 4.0 支持更多框架和语言的开发,特别是对动态语言的支持,可以让编程人员创建功能更加丰富和界面更加友好的 Web 网站。

本书内容

本书共分为 13 章,由浅入深、循序渐进地介绍了使用 ASP.NET 4.0 开发实用网站所需要掌握的技术,内容结构如下。

第 1 章 ASP.NET 4.0 概述,介绍了 Web 的开发基础、ASP.NET 4.0 开发环境 Visual Studio 2010 的安装和使用界面、ASP.NET 网址的配置,以及 IIS 的安装和配置。通过本章的学习读者会对 ASP.NET 4.0 有一个初步的认识。

第 2 章 C#入门,介绍了开发 ASP.NET 网站所使用的程序设计语言——C#。通过对 C#中关键的语法和面向对象编程知识的讲解,给读者开发 ASP.NET Web 网站打下良好的基础。

第 3 章 ASP.NET 服务器控件,介绍了 ASP.NET 4.0 中最常用的服务器控件,包括输入控件、输出控件、执行控件和面板控件等常用控件的属性、方法和使用。掌握这些控件的熟练运用,就可以设计出丰富的网页布局。

第 4 章 验证控件和用户控件,主要内容包括验证数据的方法、分类、具体数据验证控件的使用。同时对用户控件的作用和基本开发进行了详细的描述。

第 5 章 ASP.NET 常用对象,系统介绍了 ASP.NET 4.0 的常用内置对象 Page、Response、Request、Server,以及状态管理对象 Session、Cookie 和 Application。通过使用这些对象的方法和属性,可以很方便地完成众多功能。

第 6 章 访问数据库,所有网站的开发离不开数据库,本章重点介绍了 ADO.NET 对 SQL Server 关系型数据库的访问和操作,将整个操作数据库的步骤详细地介绍给读者,使读者能快速掌握访问数据库的方法。

第 7 章 数据绑定,要在网站页面呈现数据内容离不开数据绑定技术,本章从简单到复杂的数据源绑定技术进行介绍,然后详细介绍了 SqlDataSource、GridView、ListView、DetailsView 等控件的使用。

第 8 章 网站设计，分别介绍了 3 种实现网站页面设计的必备技术，即网站导航、主题和母版页。母版页对整个网站的布局风格和界面的设计统一发挥着重要的作用，主题对于页面控件的样式控制提供了极佳的帮助，而网站导航则通过对 ASP.NET 4.0 中提供的网站地图和常用的 3 种导航控件实现良好的页面导航功能。

第 9 章 LINQ 技术，LINQ 集成查询技术代替了原有的 SQL 语句，可以提供更好的完全面向对象开发的查询。本章将带领读者学习 LINQ 基础知识，掌握 LINQ 的查询语法。通过 LINQ 中的 LINQ To SQL 技术掌握对数据库数据的便捷操作。

第 10 章 Web 服务，对于流行的 Web 服务技术，本章为读者介绍了 Web 服务的基本原理、各种协议，以及在网站中如何创建、测试和调用引用 Web 服务。其中，包括使用现有 Web 服务和自定义 Web 服务。

第 11 章 ASP.NET AJAX 技术，本章介绍了风靡一时的 ASP.NET AJAX 技术。从 ASP.NET AJAX 的结构组成到核心控件的使用，最后是 AJAX CONTROL TOOLKIT 的 ASP.NET AJAX 扩展控件包的介绍，引领读者快速地进入 ASP.NET AJAX 的殿堂。

第 12 章 文件操作，对文件的操作会贯穿整个开发过程，本章将告诉读者如何使用 ASP.NET 4.0 中的各种类来对磁盘、目录、文件、文本等进行操作和读写。

第 13 章 Web 开发应用——办公自动化系统，加深读者对 ASP.NET 4.0 更好的理解。从最基本的系统分析与设计开始，确定系统的需求分析和模块划分。然后根据需求分析进行数据库和数据表的结构设计。在此基础上，创建出系统的实体类，最后是对主要模块界面的设计和实现业务逻辑的代码进行系统的介绍。通过该案例的学习，有利于读者了解一个实际项目开发的流程和所使用的技术。

本书特点

理论实际，紧密结合：本书在介绍理论知识的同时，每一章都给出了大量案例讲解，力求让读者在掌握了基础知识后，就能快速上手并举一反三。在每章末尾都配有相应的习题，方便读者课后的实践练习。

配套教学，电子教案：为了方便读者自学和教师教学，本书不仅有配套的实验指导书《Web 程序设计——ASP.NET 上机实验指导》(ISBN: 978-7-302-27923-5)，还有配套的免费课件、实例源码等。

循序渐进，步骤详细：在具体介绍 Visual Studio 2010 功能和操作时，本书提供每一步功能的具体实现，让读者很快地了解整个功能的使用方法。案例中的每一个步骤都以通俗易懂的语言进行讲述，读者只需要按照步骤操作，就可以轻松完成知识点的学习。

面向读者

本书既可以作为高等院校计算机及相关专业学生学习网站开发的技术教材，也可作为广大软件开发人员和系统架构分析设计人员自学 ASP.NET 4.0 的参考和指导用书。

鸣 谢

本书由陈作聪、王永皎、程凤娟主持编写，其中，陈作聪主编第 1~4 章，王永皎主编第 5~10 章，程凤娟主编第 11~13 章。此外，王坚宁、林丹、李辉、田芳、王建国、赵海峰、刘勇、徐超、周建军、徐兵、黄飞、林海、马建华、孙明、高峰、郑勇、刘建、李彬、彭丽、许小荣等同志在整理材料方面给予了编者很大的帮助，在此对他们表示衷心的感谢。

由于作者水平有限，书中不足之处在所难免，欢迎广大读者、同仁批评指正。

作 者

目 录

第 1 章 ASP.NET 4.0 概述.....1	
1.1 ASP.NET 4.0 框架.....1	
1.1.1 .NET 4.0 支持的语言.....1	
1.1.2 公共语言运行时.....2	
1.1.3 动态语言运行时.....2	
1.1.4 .NET 4.0 类库.....3	
1.2 网页基础知识.....3	
1.2.1 网页和服务器的交互.....3	
1.2.2 静态页面.....4	
1.2.3 动态页面.....4	
1.2.4 脚本语言.....5	
1.3 ASP.NET 应用程序.....6	
1.3.1 ASP.NET 页面与服务器交互.....6	
1.3.2 ASP.NET Web 窗体.....6	
1.3.3 后台隐藏代码页.....7	
1.3.4 ASP.NET 4.0 的新特性.....7	
1.4 建立 ASP.NET 开发和运行环境.....8	
1.4.1 安装和配置 IIS Web 服务器.....8	
1.4.2 Visual Studio 2010 开发环境.....12	
1.4.3 Visual Studio 2010 的新特性.....21	
1.5 配置 ASP.NET 4.0 应用程序.....23	
1.6 综合练习.....25	
1.7 习题.....27	
1.7.1 填空题.....27	
1.7.2 选择题.....27	
1.7.3 上机题.....28	
第 2 章 C#入门.....29	
2.1 C#代码结构.....29	
2.1.1 命名空间和类.....29	
2.1.2 Main()方法.....30	
2.1.3 语句块.....30	
2.1.4 语句终止符.....31	
2.1.5 注释.....31	
2.1.6 大小写的区别.....32	
2.2 数据类型.....32	
2.2.1 数值类型.....32	
2.2.2 布尔类型.....33	
2.2.3 结构类型.....34	
2.2.4 枚举类型.....34	
2.2.5 字符串.....35	
2.2.6 数组.....35	
2.2.7 装箱和拆箱.....38	
2.3 变量和常量.....39	
2.3.1 变量.....39	
2.3.2 常量.....39	
2.3.3 隐式局部变量.....40	
2.4 运算符和表达式.....41	
2.4.1 算术运算符.....41	
2.4.2 赋值运算符.....42	
2.4.3 关系运算符.....42	
2.4.4 逻辑运算符.....43	
2.4.5 条件运算符.....44	
2.4.6 位运算符.....44	
2.4.7 转义字符.....44	
2.5 流程控制.....45	
2.5.1 选择语句.....46	
2.5.2 循环语句.....49	
2.6 面向对象编程.....54	
2.6.1 类.....54	
2.6.2 类的成员.....55	
2.6.3 构造函数.....56	
2.6.4 继承和多态.....57	
2.6.5 事件.....59	
2.7 综合练习.....59	
2.8 习题.....61	

2.8.1 填空题	61	4.2 服务器验证控件	89
2.8.2 选择题	61	4.2.1 验证控件的分类	89
2.8.3 上机题	62	4.2.2 RequiredFieldValidator 控件	90
第 3 章 ASP.NET 服务器控件	63	4.2.3 CompareValidator 控件	92
3.1 服务器控件类	63	4.2.4 RangeValidator 控件	93
3.1.1 服务器控件基本属性	64	4.2.5 RegularExpressionValidator 控件	95
3.1.2 服务器控件的事件	64	4.2.6 CustomValidator 控件	97
3.2 执行控件	66	4.2.7 ValidationSummary 控件	98
3.2.1 普通按钮控件 Button	66	4.3 用户控件	100
3.2.2 超链接按钮控件 LinkButton	66	4.3.1 用户控件简介	101
3.2.3 图片按钮控件 ImageButton	67	4.3.2 用户控件的创建和使用	101
3.2.4 超链接文本控件 HyperLink	68	4.4 综合练习	103
3.3 输出控件	70	4.5 习题	106
3.3.1 标签控件 Label	70	4.5.1 填空题	106
3.3.2 图像控件 Image	71	4.5.2 选择题	107
3.4 输入控件	72	4.5.3 上机题	107
3.4.1 文本框控件 TextBox	72	第 5 章 ASP.NET 常用对象	109
3.4.2 复选框控件 CheckBox 和 复选框列表控件 CheckBoxList	73	5.1 Page 类	109
3.4.3 单选按钮控件 RadioButton 和单选按钮列表控件 RadioButtonList	76	5.1.1 页面的生命周期	109
3.4.4 列表框控件 ListBox	78	5.1.2 Page 类的主要属性、方法和 事件	110
3.4.5 下拉列表框控件 DropDownList	78	5.2 Request 对象	113
3.5 面板控件	81	5.2.1 Request 对象的属性和方法	113
3.6 综合练习	83	5.2.2 Request 对象的应用	113
3.7 习题	86	5.3 Response 对象	115
3.7.1 填空题	86	5.3.1 Response 对象的属性	115
3.7.2 选择题	86	5.3.2 Response 对象的应用	116
3.7.3 上机题	86	5.4 Server 对象	116
第 4 章 验证控件和用户控件	88	5.4.1 Server 对象的属性和方法	117
4.1 数据验证的两种方式	88	5.4.2 Server 对象的应用	117
4.1.1 服务器端数据验证	88	5.5 Cookie 对象	118
4.1.2 客户端数据验证	89	5.5.1 Cookie 概述	118
		5.5.2 Cookie 对象的属性和方法	119
		5.5.3 Cookie 对象的应用	119
		5.6 Session 对象	122

5.6.1 Session 概述	122	7.3.2 ListView 控件	167
5.6.2 Session 对象的属性和方法	122	7.3.3 DetailsView 控件	169
5.6.3 Session 对象的应用	123	7.4 综合练习	171
5.7 Application 对象	125	7.5 习题	173
5.8 综合练习	127	7.5.1 填空题	173
5.9 习题	131	7.5.2 选择题	173
5.9.1 填空题	131	7.5.3 上机题	174
5.9.2 选择题	131		
5.9.3 上机题	132		
第 6 章 访问数据库	133	第 8 章 网站设计	176
6.1 创建数据库	133	8.1 网站导航	176
6.2 ADO.NET 概述	135	8.1.1 网站地图	177
6.2.1 ADO.NET 简介	135	8.1.2 SiteMapDataSource 控件	179
6.2.2 ADO.NET 命名空间	136	8.1.3 导航控件	180
6.3 连接数据库	137	8.2 主题	185
6.4 获取数据	139	8.2.1 主题简介	185
6.4.1 Command 对象	139	8.2.2 主题的应用	188
6.4.2 DataReader 对象	141	8.2.3 禁用主题	190
6.5 填充数据集	144	8.3 母版页	190
6.5.1 DataAdapter 对象	144	8.3.1 母版页的定义	191
6.5.2 DataSet 对象	146	8.3.2 内容页	191
6.6 修改数据库	149	8.3.3 母版页和内容页的创建	192
6.7 综合练习	149	8.4 综合练习	193
6.8 习题	152	8.5 习题	196
6.8.1 填空题	152	8.5.1 填空题	196
6.8.2 选择题	152	8.5.2 选择题	197
6.8.3 上机题	153	8.5.3 上机题	197
第 7 章 数据绑定	155	第 9 章 LINQ 技术	199
7.1 数据绑定概述	155	9.1 LINQ 简介	199
7.1.1 绑定到简单的数据源	155	9.2 LINQ 入门	200
7.1.2 绑定到复杂的数据源	157	9.2.1 LINQ 查询步骤	200
7.2 SqlDataSource 控件	158	9.2.2 LINQ 的基本查询	201
7.2.1 SqlDataSource 控件的属性	159	9.3 LINQ 和 ADO.NET	205
7.2.2 SqlDataSource 控件的应用	160	9.3.1 LINQ to SQL 简介	206
7.3 数据服务器控件	164	9.3.2 创建对象模型	206
7.3.1 GridView 控件	164	9.3.3 LINQ 查询数据库	209
		9.3.4 LINQ 更改数据库	210
		9.4 LinqDataSource 控件	212

9.5 综合练习	213	11.3.2 CalendaeExtend 控件	259
9.6 习题	216	11.4 综合练习	261
9.6.1 填空题	216	11.5 习题	264
9.6.2 选择题	216	11.5.1 填空题	264
9.6.3 上机题	217	11.5.2 选择题	264
第 10 章 Web 服务	218	11.5.3 上机题	264
10.1 Web 服务简介	218	第 12 章 文件操作	267
10.1.1 Web 服务的概念	218	12.1 获取磁盘信息	267
10.1.2 Web 服务的基本构成	220	12.2 目录的相关操作	269
10.1.3 实现一个基本的 Web 服务	222	12.2.1 Directory 类	269
10.2 Web 服务协议	227	12.2.2 DirectoryInfo 类	274
10.2.1 SOAP	227	12.3 读写文件	280
10.2.2 WSDL	228	12.3.1 流	280
10.2.3 UDDI	230	12.3.2 FileStream 类	280
10.3 Web 服务的应用	230	12.3.3 读写文本文件	282
10.3.1 使用存在的 Web 服务	231	12.4 文件的操作	285
10.3.2 调用自定义的 Web 服务	234	12.4.1 File 类	285
10.4 综合练习	235	12.4.2 FileInfo 类	288
10.5 习题	238	12.5 综合练习	290
10.5.1 填空题	238	12.6 习题	294
10.5.2 选择题	238	12.6.1 填空题	294
10.5.3 上机题	238	12.6.2 选择题	294
第 11 章 ASP.NET AJAX 技术	240	12.6.3 上机题	294
11.1 ASP.NET AJAX 技术概述	240	第 13 章 Web 开发应用——办公自 动化系统	297
11.1.1 体系结构	240	13.1 系统分析与设计	297
11.1.2 创建 ASP.NET AJAX 程序	242	13.1.1 系统需求分析	297
11.2 ASP.NET AJAX 核心控件	243	13.1.2 系统模块设计	298
11.2.1 ScriptManager 控件	243	13.1.3 系统运行示例	299
11.2.2 UpdatePanel 控件	248	13.2 系统数据库设计	302
11.2.3 UpdateProgress 控件	252	13.2.1 数据库表设计	302
11.2.4 Timer 控件	255	13.2.2 数据库表关系	306
11.3 AJAX Control Toolkit	257	13.3 数据库管理模块——使用 LINQ 查询技术	307
11.3.1 AJAX Control Toolkit 简介	257	13.3.1 使用 LINQ 访问数据库	307
		13.3.2 实体类访问数据库	310

13.4 系统首页的设计.....	312	13.7 公共模块.....	331
13.4.1 母版页.....	312	13.7.1 界面设计	331
13.4.2 实现首页的代码.....	318	13.7.2 实现业务逻辑代码	334
13.5 系统管理模块.....	319	13.8 人事管理模块.....	337
13.5.1 系统管理模块的界面 设计.....	319	13.8.1 界面设计	337
13.5.2 实现业务逻辑代码.....	322	13.8.2 实现界面功能的业务逻辑 代码	340
13.6 个人办公模块.....	326	参考文献	343
13.6.1 界面设计.....	326		
13.6.2 实现业务逻辑代码.....	329		

第1章 ASP.NET 4.0概述

ASP.NET(Active Server Page.NET)4.0 是微软公司推出的基于.NET 4.0 框架的新一代网络编程语言,也是目前最新的 Web 技术之一。作为之前各个 ASP.NET 版本的集大成者,ASP.NET 4.0 开创了公共语言运行库和动态语言运行库相结合的编程框架,可用于在服务器上生成功能强大的 Web 应用程序。本章将介绍 ASP.NET 4.0 的相关基础知识以及如何创建其开发环境,使读者对这一强大的 Web 编程工具有一个基本的认识。

本章重点:

- 了解 ASP.NET 4.0 的基本框架
- 掌握 IIS 服务器的安装和配置
- 熟悉 Visual Studio 2010 的开发环境
- 了解 Web.config 文件的结构

1.1 ASP.NET 4.0 框架

.NET 框架是微软公司于 2002 年正式发布的新一代系统、服务和编程平台。它把原有的重点从连接到互联网的单一网站或设备转移到计算机、设备和服务群组上,从而将互联网本身作为新一代操作系统的基础。这样,用户就能够通过控制信息的传递方式、时间和内容来得到更多的服务。历时 8 年的发展,.NET 技术受到越来越多的编程人员的认可。在经历.NET 3.5 的短暂过渡之后,.NET 4.0 正式版本问世了,它的出现代表着一系列可以用来帮助我们建立丰富应用程序的技术又向前发展了一步。

1.1.1 .NET 4.0 支持的语言

.NET 4.0 框架支持多种语言,包括 C#、VB、J#和 C++等,本书在后台使用的语言主要是 C#。C#是在.NET 1.0 中开始出现的一种新语言,在语法上,它与 Java 和 C++比较相似。实际上 C#是微软整合了 Java 和 C++的优点而开发出来的一种语言,也是微软对抗 Java 平台的一个有效工具。

在被执行之前,所有.NET 4.0 语言都会被编译成为一种低级别的语言,这种语言就是中间语言(Intermediate Language, IL)。CLR 之所以支持很多种语言,就是因为这些语言在运行之前被编译成了中间语言。正是因为所有的.NET 语言都是建立在中间语言之上,所以 VB 和 C#具有相同的特性和行为。因此一个使用 C#编写的 Web 页面也可以使用 VB 编写

的组件, 同样使用 VB 编写的 Web 页面也可以使用 C# 编写的组件。

.NET 4.0 框架提供了一个公共语言规范(Common Language Specification, CLS)以保证这些语言之间的兼容性。只要遵循 CLS, 任何利用某一种 .NET 语言编写的组件都可以被其他语言所引用。CLS 的一个重要部分是公共类型系统(Common Type System, CTS), CTS 定义了诸如数字、字符串和数组等数据类型的规则, 这样它们就能为所有的 .NET 语言所共享。CLS 还定义了诸如类、方法、实践等对象成分。然而事实上, 基于 .NET 进行程序开发的程序员却没有必要考虑 CLS 是如何工作的, 因为这一切都由 .NET 4.0 平台自动完成。CLR 只执行中间语言代码, 然后把它们进一步编译成为机器语言代码以能够使当前平台所执行。

1.1.2 公共语言运行时

公共语言运行时(Common Language Runtime, CLR)是指用 .NET 4.0 语言编写的代码公共运行环境。它既是 .NET 框架的基础, 也是实现 .NET 跨平台、跨语言、代码安全等核心特性的关键。公共语言运行时就像一个执行程序时管理代码的代理, 以跨语言集成、自描述组件、简单配制和版本化及集成安全服务为特点, 提供核心服务(如内存管理、线程管理和远程处理)。

公共语言运行时管理的 .NET 4.0 中的代码, 称为受托管代码。它们包含了有关代码的信息, 例如代码中定义的类、方法和变量。受托管代码中所包含的信息称为元数据。公共语言运行时使用元数据来安全地执行代码程序。除了安全地执行程序以外, 受托管代码的目的在于 CLR 服务。这些服务包括查找和加载类, 以及与现有的动态链接库(Dynamic Link Library, DLL)代码和组件对象之间的相互操作。

公共语言运行时遵循公共语言架构的标准, 可以使 C++、C#、Visual Basic 以及 JScript 等多种语言能够深度集成。

1.1.3 动态语言运行时

.NET 4.0 框架中最令人激动的新特性是动态语言运行时(Dynamic Language Runtime, DLR)。就像公共语言运行时(CLR)为静态型语言(如 C# 和 Visual Basic)提供了通用平台一样, 动态语言运行时(DLR)为动态型语言(如 JavaScript、Ruby、Python)甚至 COM 组件等提供了通用平台, 这代表 .NET 4.0 框架在互操作性方面向前迈进了一大步。

动态语言运行时(DLR)是一种运行时环境, 它将一组适用于动态语言的服务添加到公共语言运行时。借助于动态语言运行时, 可以更轻松地开发要在 .NET 4.0 框架上运行的动态语言, 而且向静态类型化语言添加动态功能也会变得更容易。

动态语言运行时的目的是允许动态语言系统在 .NET 框架上运行, 并为动态语言提供 .NET 互操作性。在 Visual Studio 2010 中, 动态语言运行时将动态对象引入到 C# 和 Visual Basic 中, 以便这些语言能够支持动态行为, 并且可以与动态语言进行互操作, 同

时动态语言运行时还可帮助用户创建支持动态操作的库。

与公共语言运行时类似, 动态语言运行时是 .NET 4.0 框架的一部分, 并随 .NET Framework 4.0 和 Visual Studio 2010 安装包一起提供。

动态语言运行时通过在调用站点中使用联编程序, 不仅可以与 .NET Framework 通信, 还可以与其他基础结构和服务(包括 Silverlight 和 COM)通信。联编程序将封装语言的语义, 并指定如何使用表达式在调用站点中执行操作。这样, 使用动态语言运行时的动态和静态类型化语言就能够共享类库, 并获得对动态语言运行时支持的所有技术的访问权。

1.1.4 .NET 4.0 类库

.NET 4.0 框架的另一个主要组件是类库, 它是一个综合性的面向对象的可重用类型集合, 例如 ADO.NET、ASP.NET 等。.NET 基类库位于公共语言运行库的上层, 与 .NET Framework 紧密集成在一起, 可被 .NET 支持的任何语言所使用。这也就是为什么 ASP.NET 中可以使用 C#、VB.NET、VC.NET 等语言进行开发。.NET 类库非常丰富, 提供数据库访问、XML、网络通信、线程、图形图像、安全、加密等多种功能服务。类库中的基类提供了标准的功能, 如输入输出、字符串操作、安全管理、网络通信、线程管理、文本管理和用户界面设计功能。这些类库使得开发人员更容易地建立应用程序和网络服务, 从而提高开发效率。

1.2 网页基础知识

要开发一个网站, 首先要了解组成网站的最基本的元素——网页。本节就来了解一下网页的基础知识, 包括网页和服务器的交互过程、静态和动态网页以及脚本语言。

1.2.1 网页和服务器的交互

通过互联网浏览网页时, 用户会自动与网页服务器建立连接。用户提交信息资源的过程称为向服务器发出请求。通过服务器解释信息资源来定位对应的页面, 并传送回代码来创建页面, 这个过程称为对浏览器的响应。浏览器接受来自于网页服务器的代码, 并将它编译成可视页面。在这样的交互过程中, 浏览器称为“客户机”或者“客户端”, 整个交互的过程则称为“客户机/服务器”的通信过程。

“客户机/服务器”概括了任务的分布来描述网页的工作方式。服务器(Web 服务器)存储、解释和分布数据。客户机(浏览器)访问服务器以得到这些数据。客户机和服务器使用 HTTP 协议通过 Internet 进行交互。HTTP 协议又叫做超文本传输协议, 是一个客户机和服务器端请求和应答的标准。浏览器网页时, 浏览器通过 HTTP 协议与服务器交换信息。

1.2.2 静态页面

早期的网站发布的是静态网页，主要由 HTML 语言组成，没有其他可以执行的程序代码。静态页面一经制成，内容就不会再改变，不管何时何人访问，显示的都是一样的内容，如果要修改有关内容，就必须修改源代码，然后重新上传到服务器上。静态页面虽然包含文字和图片，但这些内容却需要在服务器端以手工的方式来变换，因此很难把它们描述为 Web 程序。下面是使用 HTML 语言编写的一个简单静态网页代码：

```
<html>
  <head>
    <title>这是一个静态网页</title>
  </head>
  <body>
    <h1>这是一个静态网页</h1>
    <p>静态网页使用 HTML 语言编写</p>
  </body>
</html>
```

代码说明：该程序包含一个标题和一句文字。其中标题包含在标记<h1>和</h1>之间，文字包含在标记<p>和</p>之间。图 1-1 显示了该静态网页文件被浏览器解析后的结果。

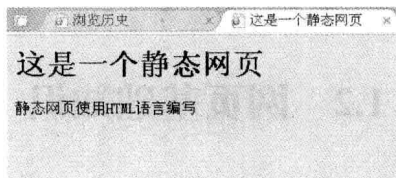


图 1-1 静态网页

HTML 是互联网的描述语言，基本的 HTML 语言包含由 HTML 标记格式化的文本和图像内容。文本是 HTML 要显示的内容，标记则告诉浏览器如何显示这些内容，它定义了不同层次的标题、段落、链接、斜体格式化、横向线等。HTML 文件的后缀可以是.htm，也可以是.html。

1.2.3 动态页面

动态页面不仅含有 HTML 标记，而且含有可以执行的程序代码，动态页面能够根据不同的输入和请求动态生成返回的页面，例如常见的 BBS、留言板、聊天室等就是用动态网页来实现的。动态网页的使用非常灵活，功能强大。

真正意义上包含动态页面的 Web 程序，直到 HTML 2.0 中 HTML 表单引入时才开始出现。在一个 HTML 表单中，所有的控制都放置在<form>和</form>标记中。当用户在客户端单击“提交”按钮后，网页上的所有内容就以字符串的形式发送到服务器端，服务器

端的处理程序根据事先设置好的标准来响应客户的请求。下面是一个由 HTML 表单构成的动态页面代码。

```
<html>
  <head>
    <title>这是一个动态网页</title>
  </head>
  <body>
    <form>
      <h3>请选择您喜欢的水果? </h3>
      <p>请做出选择: </p>
      <input type="checkbox" />苹果<br/>
      <input type="checkbox" />香蕉<br/>
      <input type="checkbox" />芒果<br/>
      <input type="checkbox" />提子<br/>
      <input type="submit" value="提交">
    </form>
  </body>
</html>
```

代码说明: 该程序由 HTML 表单组成, 包括一个标题、四个复选框和提交按钮, 这些内容和标记均被包含在表单标记之间。该网页运行效果如图 1-2 所示。

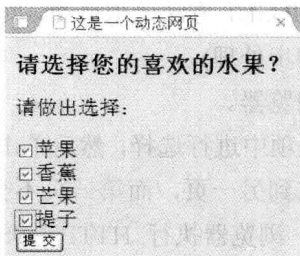


图 1-2 动态页面

现在, 尽管动态 ASP.NET 页面已经比较流行, 但 HTML 表单仍然是这些页面的基本组成元素, 所不同的是构成 ASP.NET 页面的 HTML 表单控件运行在服务器端, 所以读者必须要掌握最基本的 HTML 表单, 以便能够更好地使用 ASP.NET 平台进行程序开发。

1.2.4 脚本语言

在网页的发展过程中出现了很多优秀脚本语言, 如 ASP、JSP、PHP 等。脚本语言确实简化了 Web 程序的开发, 但其使用起来也有很大的缺点。首先, 它的代码和 HTML 标记杂乱地堆砌在一起, 显得很混乱, 非常不方便开发和维护, 所以当 ASP.NET 的代码和 HTML 标记分离后, 使用以往一些脚本语言的 Web 开发人员都有一种耳目一新的感觉; 其次, 脚本语言的编程思想不符合当前流行的面向对象编程思想。基于以上原因, 脚本语言必将会被其他更高级语言(如 ASP.NET 和 Java 等)所代替。

1.3 ASP.NET 应用程序

ASP.NET 应用程序是一系列资源和配置的组合,这些资源和配置只在同一个应用程序内共享,而其他应用程序则不能享用这些资源和配置,尽管它们发布在同一台服务器上。就技术而言,每个 ASP.NET 应用程序都运行在一个单独的应用程序域,应用程序域是内存中的独立区域,这样可以确保在同一台服务器上的应用程序不会相互干扰,不至于因为其中一个应用程序发生错误就影响到其他应用程序的正常进行。同样,应用程序域限制一个应用程序中的 Web 页面访问其他的应用程序的存储信息。每个应用程序单独运行,具有自己的存储、应用和会话数据。

ASP.NET 应用程序的标准定义是:文件、页面、处理器、模块和可执行代码的组合,并且它们能够从服务器上一个虚拟目录中被引用。换句话说,虚拟目录是界定应用程序的基本组织结构。

1.3.1 ASP.NET 页面与服务器交互

ASP.NET 页面作为代码在服务器上运行。在用户单击按钮(或者当用户选中复选框或与页面中的其他控件交互)时提交页面到服务器。每次页面都会回发,以便它可以再次运行其服务器代码,然后向用户呈现其自身的新版本。传递 Web 页面的具体过程如下:

- 用户请求页面。使用 HTTP GET 方法请求页面,页面第一次运行,执行初步处理(如果已通过编程让它执行初步处理)。
- 页面将标记动态呈现到浏览器。
- 用户输入信息或从可用选项中进行选择,然后单击按钮。如果用户单击链接而不是按钮,页面可能仅仅定位到另一页,而第一页不会被进一步处理。
- 页面发送到 Web 服务器。浏览器执行 HTTP POST 方法,该方法在 ASP.NET 中称为“回发”。更明确地说,页面发送回其自身。例如,如果用户正在使用 Default.aspx 页面,则单击该页上的某个按钮可以将该页发送回服务器,发送的目标则是 Default.aspx。
- 在 Web 服务器上,该页再次运行。并且可在页面上使用用户输入或选择的信息。
- 页面执行通过编程所要实行的操作。
- 页面将其自身呈现回浏览器。

只要用户在该页面中工作,此循环就会继续。用户每次单击按钮时,页面中的信息会发送到 Web 服务器,然后该页面再次运行。每个循环称为一次“往返行程”。由于页面处理发生在 Web 服务器上,因此页面可以执行的每个操作都需要一次到服务器的往返行程。

1.3.2 ASP.NET Web 窗体

在 ASP.NET 中,发送到客户端浏览器中的网页是经过.NET 框架中的基类动态生成的。