

第一部分 基础知识

第 1 章 ASP.NET 2.0 概述	2
1.1 .NET 技术发展回顾	2
1.2 ASP.NET 2.0 设计目标	3
1.3 ASP.NET 2.0 新特性	3
1.3.1 新增服务器控件	4
1.3.2 母版页	7
1.3.3 主题和皮肤	8
1.3.4 个性化用户配置	8
1.3.5 成员资格和角色管理	9
1.3.6 配置和管理工具	10
1.4 小结	11
第 2 章 Visual Studio 2005 集成开发环境	12
2.1 安装 Visual Studio 2005	12
2.2 Visual Studio 2005 新特性	15
2.3 针对 ASP.NET 应用开发的新特性	16
2.4 常用功能窗口介绍	17
2.4.1 主窗口	17
2.4.2 文档窗口	18
2.4.3 工具箱窗口	19
2.4.4 解决方案资源管理器窗口	20
2.4.5 服务器资源管理器窗口	20
2.4.6 属性窗口	21
2.5 配置设置	21
2.5.1 选项窗口	21
2.5.2 导入和导出设置向导	23
2.6 帮助系统	26

2.7	类设计器	28
2.8	多种测试功能	29
2.9	小结	31
第 3 章	创建 ASP.NET 2.0 应用程序	32
3.1	创建 Web 站点	32
3.2	编写 ASP.NET 2.0 应用程序	37
3.3	编译和运行应用程序	39
3.4	部署应用程序	43
3.5	小结	44
第 4 章	C# 2.0 程序设计基础	45
4.1	C#语言概述	45
4.2	数据类型	47
4.2.1	值类型	47
4.2.2	引用类型	50
4.2.3	装箱和拆箱	53
4.3	变量和常量	54
4.3.1	变量	54
4.3.2	常量	56
4.4	流程控制	56
4.4.1	条件语句	56
4.4.2	循环语句	58
4.4.3	跳转语句	61
4.4.4	异常处理	63
4.5	面向对象的特性	65
4.5.1	类	65
4.5.2	属性	66
4.5.3	方法	67
4.5.4	事件	67
4.6	C# 2.0 新特性	69
4.6.1	泛型	69
4.6.2	匿名方法	76
4.6.3	迭代器	82
4.6.4	局部类	88
4.7	编码规范	90
4.7.1	代码格式	90
4.7.2	注释规范	90
4.7.3	变量命名规范	92
4.7.4	常量命名规范	93
4.7.5	类命名规范	93

4.7.6 接口命名规范	94
4.7.7 方法命名规范	94
4.7.8 名字空间命名规范	94
4.7.9 资源命名规范	94
4.7.10 程序版本号确定规范	94
4.8 小结	94

第二部分 用户界面

第5章 母版页	96
5.1 母版页概述	96
5.1.1 母版页基础知识	97
5.1.2 母版页运行机制	99
5.1.3 母版页和内容页事件顺序	100
5.1.4 母版页的优点	101
5.2 创建母版页	102
5.3 创建内容页	105
5.4 嵌套母版页	107
5.5 访问母版页控件和属性	112
5.5.1 使用 FindControl 方法获取母版页控件引用	112
5.5.2 使用 MasterType 指令获取母版页控件引用	115
5.5.3 访问母版页简单自定义属性	118
5.6 动态加载母版页	121
5.7 母版页应用范围	127
5.8 母版页缓存设置	128
5.9 针对不同客户端的处理	129
5.10 小结	129
第6章 主题和皮肤	130
6.1 主题概述	130
6.1.1 组成元素	130
6.1.2 文件存储和组织方式	131
6.1.3 全局主题和应用程序主题	132
6.2 创建主题	132
6.2.1 创建皮肤文件	132
6.2.2 为主题添加 CSS 文件	135
6.2.3 在主题中使用图片	137
6.3 应用主题	139
6.3.1 指定和禁用主题	139
6.3.2 动态加载主题	141
6.4 小结	144

第 7 章 站点导航控件	145
7.1 站点地图概述	145
7.2 SiteMapPath 控件	147
7.2.1 SiteMapPath 控件简介	147
7.2.2 常用属性列表	147
7.2.3 常用事件列表	149
7.2.4 典型应用 1——基本应用	149
7.2.5 典型应用 2——设置自定义模板	153
7.3 TreeView 控件	155
7.3.1 TreeView 控件简介	156
7.3.2 常用属性列表	157
7.3.3 常用事件列表	158
7.3.4 典型应用 1——基本应用	158
7.3.5 典型应用 2——绑定 XML 文件	161
7.4 Menu 控件	167
7.4.1 Menu 控件简介	168
7.4.2 常用属性列表	169
7.4.3 常用事件列表	171
7.4.4 典型应用 1——基本应用	171
7.4.5 典型应用 2——绑定 XML 文件	175
7.5 小结	181
第 8 章 其他服务器控件	182
8.1 BulletedList 控件	182
8.1.1 BulletedList 控件简介	182
8.1.2 典型应用	184
8.2 HiddenField 控件	189
8.2.1 HiddenField 控件简介	189
8.2.2 典型应用	190
8.3 FileUpload 控件	192
8.3.1 FileUpload 控件简介	192
8.3.2 典型应用	194
8.4 ImageMap 控件	196
8.4.1 ImageMap 控件简介	197
8.4.2 典型应用	198
8.5 MultiView 和 View 控件	201
8.5.1 MultiView 和 View 控件简介	201
8.5.2 典型应用	202
8.6 Wizard 控件	206
8.6.1 Wizard 控件简介	206
8.6.2 典型应用	212

8.7 小结	218
--------------	-----

第三部分 数据访问

第 9 章 数据访问技术	220
9.1 数据访问概述	220
9.2 数据源控件简介	222
9.3 数据绑定控件简介	224
9.4 小结	225
第 10 章 数据源控件	226
10.1 SqlDataSource 控件	226
10.1.1 SqlDataSource 控件简介	226
10.1.2 常用属性列表	228
10.1.3 常用方法列表	230
10.1.4 常用事件列表	231
10.1.5 典型应用 1——连接数据库	231
10.1.6 典型应用 2——属性设置	242
10.1.7 典型应用 3——连接多种数据源	247
10.2 AccessDataSource 控件	252
10.2.1 AccessDataSource 控件简介	253
10.2.2 常用属性列表	254
10.2.3 常用方法列表	257
10.2.4 常用事件列表	257
10.2.5 典型应用——访问 Access 数据库文件	257
10.3 XmlDataSource 控件	267
10.3.1 XmlDataSource 控件简介	267
10.3.2 常用属性列表	268
10.3.3 常用方法列表	269
10.3.4 常用事件列表	270
10.3.5 典型应用——显示 RSS 数据	270
10.4 SiteMapDataSource 控件	278
10.4.1 SiteMapDataSource 控件简介	279
10.4.2 常用属性列表	280
10.4.3 常用方法列表	281
10.4.4 典型应用 1——属性设置	281
10.4.5 典型应用 2——绑定普通数据绑定控件	284
10.5 ObjectDataSource 控件	285
10.5.1 N 层应用程序概述	286
10.5.2 ObjectDataSource 控件简介	287
10.5.3 常用属性列表	288
10.5.4 常用方法列表	291

10.5.5 常用事件列表	291
10.5.6 典型应用 1——绑定数据访问层	292
10.5.7 典型应用 2——绑定业务逻辑层	305
10.5.8 典型应用 3——绑定类型化 DataSet	322
10.6 小结	329
第 11 章 数据绑定控件	330
11.1 GridView 控件	330
11.1.1 GridView 控件简介	330
11.1.2 常用属性列表	335
11.1.3 常用方法列表	339
11.1.4 常用事件列表	339
11.1.5 典型应用——使用 GridView 显示和编辑数据	340
11.2 DetailsView 控件	345
11.2.1 DetailsView 控件简介	345
11.2.2 常用属性列表	347
11.2.3 常用方法列表	351
11.2.4 常用事件列表	351
11.2.5 典型应用——使用 DetailsView 控件显示和处理数据	352
11.3 FormView 控件	358
11.3.1 FormView 控件简介	358
11.3.2 常用属性列表	361
11.3.3 常用方法列表	363
11.3.4 常用事件列表	364
11.3.5 典型应用——使用 FormView 控件显示和处理数据	364
11.4 小结	373
第 12 章 数据缓存	374
12.1 ASP.NET 2.0 缓存概述	374
12.2 页面输出缓存	376
12.2.1 使用 @OutputCache 指令	377
12.2.2 使用页面输出缓存 API	378
12.2.3 页面输出缓存应用	380
12.3 页面部分缓存	381
12.3.1 使用 @OutputCache 指令	382
12.3.2 使用 PartialCaching Attribute 类	383
12.3.3 使用 ControlCachePolicy 类	384
12.3.4 实现缓存后替换	386
12.4 应用程序数据缓存	391
12.4.1 添加应用程序数据缓存对象	391
12.4.2 检索应用程序数据缓存对象	394
12.4.3 移除应用程序数据缓存对象	395

12.4.4 实现应用程序数据缓存	395
12.5 缓存依赖	399
12.5.1 缓存依赖概述	399
12.5.2 实现自定义缓存依赖	400
12.5.3 实现聚合缓存依赖	408
12.5.4 实现 SQL 数据缓存依赖	409
12.6 小结	422

第四部分 应用服务

第 13 章 个性化用户配置	424
13.1 个性化功能简介	424
13.1.1 个性化服务的三大步骤	425
13.1.2 实现个性化服务的三大功能	425
13.2 个性化用户配置概述	426
13.2.1 个性化用户配置的功能	426
13.2.2 <profile>配置节声明	427
13.2.3 与个性化用户配置功能有关的 API	430
13.2.4 SQL Server 数据库配置	433
13.3 为匿名用户实现简单数据类型的个性化用户配置	439
13.4 为注册用户实现复杂数据类型的个性化用户配置	444
13.4.1 实现概述	444
13.4.2 实现用户登录和验证	445
13.4.3 实现购物车	448
13.4.4 应用个性化用户配置功能	451
13.5 实现匿名用户向注册用户迁移	456
13.6 实现自定义 ProfileBase 类	460
13.7 实现自定义用户配置提供程序	465
13.7.1 实现概述	465
13.7.2 步骤一——创建 Access 数据库	468
13.7.3 步骤二——设置事件日志访问	470
13.7.4 步骤三——实现自定义用户配置提供程序	471
13.7.5 步骤四——配置 Web.config 文件	492
13.7.6 步骤五——示例应用	494
13.8 小结	497
第 14 章 Web 部件	498
14.1 Web 部件概述	498
14.1.1 什么是 Web 部件	498
14.1.2 Web 部件具体功能和应用方案	500
14.1.3 Web 部件基本控件	501
14.2 创建 WebPart 控件	503

14.2.1	WebPartManager 控件简介	503
14.2.2	WebPartZone 控件和 WebPart 控件简介	507
14.2.3	创建 WebPart 控件	510
14.3	编辑 WebPart 控件	520
14.3.1	EditorZone 控件简介	520
14.3.2	AppearanceEditorPart 控件简介	521
14.3.3	BehaviorEditorPart 控件简介	522
14.3.4	LayoutEditorPart 控件简介	525
14.3.5	PropertyGridEditorPart 控件简介	526
14.3.6	实现编辑 WebPart 控件	527
14.4	管理 WebPart 控件	536
14.4.1	CatalogZone 控件简介	536
14.4.2	DeclarativeCatalogPart 控件简介	536
14.4.3	PageCatalogPart 控件简介	538
14.4.4	ImportCatalogPart 控件简介	539
14.4.5	实现管理 WebPart 控件	540
14.5	WebPart 控件通信	546
14.5.1	WebPart 控件通信概述	546
14.5.2	实现基于静态连接的 WebPart 通信	550
14.5.3	实现基于动态连接的 WebPart 通信	558
14.6	Web 部件个性化	560
14.6.1	Web 部件个性化概述	560
14.6.2	实现自定义 Web 部件个性化提供程序	564
14.7	小结	580
第 15 章	成员资格和角色管理	581
15.1	身份验证和授权	581
15.1.1	身份验证概述	582
15.1.2	Forms 验证概述	583
15.1.3	FormsAuthentication 类	589
15.1.4	用户授权概述	591
15.2	成员资格管理	593
15.2.1	成员资格管理概述	593
15.2.2	Membership 类	600
15.2.3	MembershipUser 类	603
15.2.4	实现基本成员资格管理功能	605
15.2.5	实现自定义成员资格提供程序	622
15.3	角色管理	634
15.3.1	角色管理概述	634
15.3.2	Roles 类	640
15.3.3	实现基本角色管理功能	643
15.3.4	实现自定义角色管理提供程序	655

15.4 小结	666
第 16 章 登录系列控件	668
16.1 登录系列控件概述	668
16.2 Login 控件	670
16.3 LoginName 控件	674
16.4 LoginView 控件	674
16.5 LoginStatus 控件	677
16.6 PasswordRecovery 控件	678
16.7 ChangePassword 控件	685
16.8 CreateUserWizard 控件	692
16.9 综合应用示例	700
16.9.1 实现用户登录	700
16.9.2 创建新用户	701
16.9.3 显示状态和修改密码	703
16.9.4 重置密码	704
16.9.5 配置 Web.config 文件	705
16.10 小结	705
第 17 章 配置与管理工具	706
17.1 ASP.NET MMC 管理单元	706
17.1.1 ASP.NET MMC 管理单元简介	706
17.1.2 “常规”选项卡	708
17.1.3 “自定义错误”选项卡	710
17.1.4 “授权”选项卡	712
17.1.5 “身份验证”选项卡	714
17.1.6 “应用程序”选项卡	716
17.1.7 “状态管理”选项卡	717
17.1.8 “位置”选项卡	718
17.2 Web 网站管理工具	720
17.2.1 Web 网站管理工具简介	720
17.2.2 “安全”选项卡	722
17.2.3 “应用程序”选项卡	729
17.2.4 “提供程序”选项卡	734
17.3 小结	735

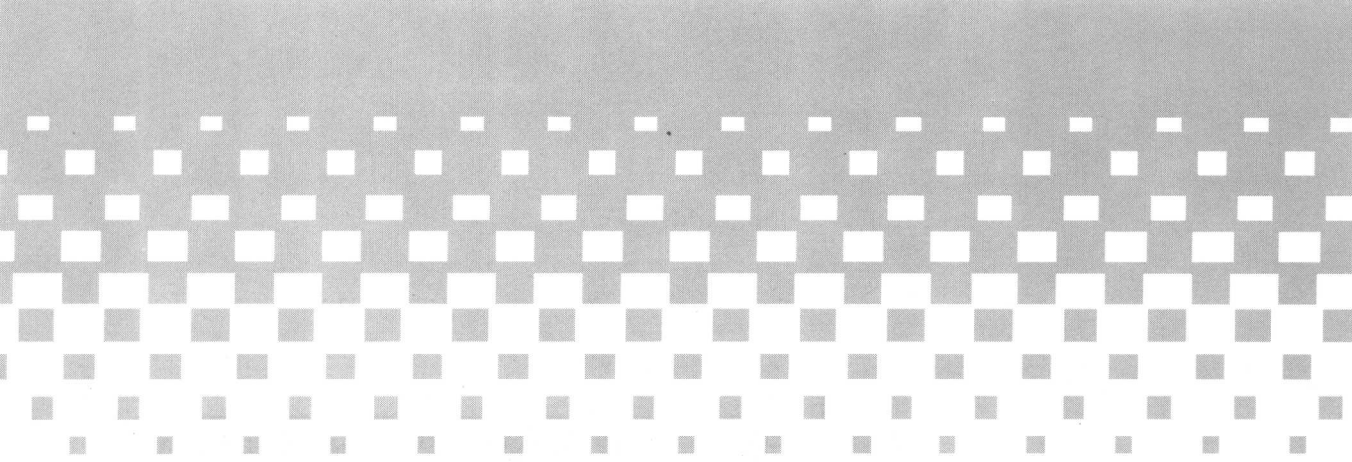
第五部分 实例分析

第 18 章 PetShop 4.0 实例分析	738
18.1 应用程序概述	738
18.1.1 安装过程	738
18.1.2 功能和用户界面简介	740

18.1.3	解决方案和体系架构概述	745
18.2	数据访问层	749
18.2.1	应用程序数据模型	749
18.2.2	数据库连接字符串加密/解密	752
18.2.3	数据访问辅助项目 DBUtility 分析	755
18.2.4	部分业务逻辑所需的数据访问项目分析	758
18.2.5	自定义成员资格提供程序项目 Membership 分析	767
18.2.6	自定义用户配置提供程序项目分析	769
18.2.7	消息队列访问项目分析	778
18.3	业务逻辑层	784
18.3.1	业务实体项目 Model 分析	784
18.3.2	业务逻辑项目 BLL 分析	787
18.3.3	订单策略项目 IBLLStrategy 分析	789
18.4	表示层	796
18.4.1	订单处理控制台应用程序项目 OrderProcessor 分析	796
18.4.2	数据缓存依赖项目分析	803
18.4.3	Web 项目分析	810
18.5	小结	812

第六部分 附 录

附录 A	SQL 语言实用参考	814
附录 B	ASP.NET 技术推荐参考资源	817



»»» 第一部分

基 础 知 识

- 第 1 章 ASP.NET 2.0 概述
- 第 2 章 Visual Studio 2005 集成开发环境
- 第 3 章 创建 ASP.NET 2.0 应用程序
- 第 4 章 C# 2.0 程序设计基础

第1章 ASP.NET 2.0 概述

本章要点

- .NET 技术发展回顾
- ASP.NET 2.0 设计目标
- ASP.NET 2.0 新特性

欢迎来到 ASP.NET 2.0 的世界!

从本章开始,将向读者详细介绍 ASP.NET 2.0 技术。相对于 ASP.NET 1.x 而言,ASP.NET 2.0 增加了很多新技术特性。虽然这些新特性,能够帮助开发人员明显提高工作效率,但同时也带来了学习方面的困难。例如,每个特性都包括众多概念、控件、应用方法等,这些内容对于开发人员,尤其是从未接触过 ASP.NET 技术的人员来讲都是全新的。为了帮助读者对将要学习的内容,建立宏观概念,本章首先介绍 ASP.NET 技术的发展历史,然后,说明 ASP.NET 2.0 的设计目标,最后,对 ASP.NET 2.0 的新特性进行概括性介绍。

1.1 .NET 技术发展回顾

从 2000 年 .NET 技术崭露头脚,到 2005 年末推出 .NET 2.0,微软公司为推广 .NET 技术可以说是 不遗余力。下面简单介绍一下 .NET 技术的发展历程。

- 2000 年 6 月,微软公司总裁比尔·盖茨先生在一次名为“论坛 2000”的会议上发表演讲,描绘了 .NET 的美景。
- 2002 年 1 月,微软公司公布 .NET Framework 1.0 正式版。与此同时,Visual Studio.NET 2002 也同步发行。
- 2003 年 4 月 23 日,微软公司推出 .NET Framework 1.1 和 Visual Studio.NET 2003。这些重量级的产品都是针对 .NET 1.0 的升级版本。
- 2004 年 6 月,在 TechEd Europe 会议上,微软发布 .NET Framework 2.0 Beta1 和 Visual Studio 2005 Beta1。同时,还发布多个精简版(Express Edition),其中包括 Visual Web Developer 2005、Visual Basic 2005、Visual C# 2005 和 SQL Server 2005 Express Edition 等。
- 2005 年 4 月,微软公司发布 Visual Studio 2005 Beta2 测试版。
- 2005 年 11 月,微软公司发布 Visual Studio 2005 和 SQL Server 2005 正式版。

在 .NET 1.0 发布后,也就是 2002 年及其随后一两年的时间内,.NET 技术一直处在发展初期。

虽然微软公司不遗余力的宣传.NET 技术,但是,使广大开发人员尤其是软件开发商接受.NET,则需要一个比较漫长的过程。.NET 1.1 发布后,学习和使用.NET 技术的热潮开始不断涌现。目前,微软发布的.NET 最新版本是 2.0。.NET 2.0 的发布是.NET 技术走向成熟的标志。尤其是用于 Web 应用程序开发的核心技术,ASP.NET 2.0 更是万众瞩目,不断吸引着越来越多的目光。为了使读者对 ASP.NET 2.0 建立初步概念,首先介绍 ASP.NET 2.0 的设计目标。

1.2 ASP.NET 2.0 设计目标

ASP.NET 2.0 的开发代号是“ASP.NET Whidbey”。Whidbey 是 Visual Studio 2005 的开发代号。由于 ASP.NET 应用程序通常使用 Visual Studio 2005 创建,因此,ASP.NET 2.0 被称为“ASP.NET Whidbey”也确实恰如其分。

在 ASP.NET 2.0 设计过程中,微软公司深入市场,针对大量开发人员和软件公司的需求,进行了卓有成效的研究。这些艰苦细致的工作,为 ASP.NET 2.0 的开发打下了良好的基础。同时,也为 ASP.NET 2.0 这一产品设定了宏伟目标。ASP.NET 2.0 设计目标的核心用一个词来形容——简化。围绕这个核心,建立了以下几个主要设计目标。

● 提高开发效率

设计目标是将应用程序代码行数减少约 70%,以提高开发效率,进而开发人员可以将更多的注意力转移到如何实现业务逻辑上来,彻底改变过去那种需要编写很多重复性代码的状况。对于部分开发工作,尽量做到在不写或者少写代码的情况下,也可完成任务。

● 简化管理和维护工作

在 ASP.NET 1.x 技术中,Web 应用程序的管理和维护主要通过人工方式,即对 XML 配置文件(例如 machine.config 和 web.config)进行处理来完成。这种方式易于出错。在 ASP.NET 2.0 中,目标是创建图形化管理工具,简化管理和维护工作。

● 提高性能和可伸缩性

在 ASP.NET 1.x 中,应用程序性能和可伸缩性一直是两个饱受争议的话题。然而,无论结果如何,所有人共同的观点是必须对性能和可伸缩性进行改进。ASP.NET 2.0 在以上两个方面进行了重大改进,主要体现在优化内部处理机制、扩展高速缓存功能以及增加对 64 位服务器支持等方面。

● 扩展对于不同终端设备的支持能力

优化 ASP.NET 2.0 技术,提高对于浏览器、PDA、移动电话等终端设备的支持能力,并且增加对各种未来终端设备的支持。另外,通过扩展控件功能,使得同一控件能够输出支持多种设备的代码。

ASP.NET 2.0 的设计目标相当宏伟壮丽,以上只是简单介绍了其中的几个主要目标。围绕设计目标,ASP.NET 2.0 新增了多个技术特性。下一节将对 ASP.NET 2.0 新特性进行概括性介绍。

1.3 ASP.NET 2.0 新特性

ASP.NET 技术从 1.0 版本升级到 1.1 变化不是很大。然而,从 ASP.NET 1.x 升级到 2.0,却不是件轻而易举的事情。ASP.NET 2.0 技术增加了大量方便、实用的新特性。图 1-1 所示列举了 ASP.NET 2.0 的主要技术特性。

如图 1-1 所示,ASP.NET 2.0 主要提供控件、页面框架、服务与 APIs 等 3 个方面的技术特性。



图 1-1 ASP.NET 2.0 技术特性

● 控件

在 ASP.NET 1.x 时代，由于内置服务器控件数量有限、功能覆盖面窄，因此，开发人员怨声载道。为了消除这个技术软肋，ASP.NET 2.0 做出了突破性改进。在 ASP.NET 2.0 中，新增数十个服务器控件。根据控件功能特征，可分为：数据控件（包括数据源控件和数据绑定控件）、登录系列控件、Web 部件、网站导航控件和其他控件等。这些服务器控件对于提高工作效率，降低开发成本有着重要作用。

● 页面框架

在页面框架方面，ASP.NET 2.0 同样做出了重大改进，主要体现在以下 4 项技术特性：母版页、主题和皮肤、编译机制和本地化。母版页特性在大型 Web 站点中应用广泛，它能够在一定程度上代替框架页功能，从而迅速建立具有统一布局和外观的页面，并且有效降低维护人员的工作强度。主题和皮肤是 ASP.NET 2.0 的一项新增功能，使用此功能可以一次性设置一种类型的服务器控件外观样式。该外观可方便地应用于单个页面、整个 Web 应用程序，甚至整个服务器。在应用程序编译方面，ASP.NET 2.0 提供了几种替代方案，从而扩大了编译的选择范围，开发人员可以根据特定需要进行选择。另外，ASP.NET 2.0 在本地化方面也做出了一些改进。

● 服务与 APIs

这个部分包括的新特性在 ASP.NET 2.0 中占有较大比例，主要包括成员资格管理、角色管理、用户配置、数据缓存、配置与管理工具等。利用这些功能，可以快速实现用户和角色管理、个性化用户配置、数据缓存、应用程序可视化配置等任务。以成员资格和角色管理功能为例，在 ASP.NET 1.x 时代，实现这两项功能是比较繁琐和困难的；而在 ASP.NET 2.0 中不仅提供默认数据库表，而且只需调用几个简单的方法，就能够轻松地实现成员和角色管理功能。

注意：在日常开发工作中本地化功能应用较少，因此，本书未涉及相关内容。

1.3.1 新增服务器控件

在 ASP.NET 1.x 时代，有很多开发人员对其中的内置服务器控件提出批评。主要的观点是

ASP.NET 1.x 提供的控件数量太少, 难以满足日益增长的开发需求, 无法更快更好地开发应用程序。为了弥补 ASP.NET 1.x 的不足, ASP.NET 2.0 新增了数十个服务器控件。根据控件功能, 可以分为站点导航控件、数据控件 (包括数据源控件和数据绑定控件)、登录系列控件、Web 部件和其他服务器控件等。熟练使用这些服务器控件对于提供工作效率、减低开发成本有着重要意义。

1. 站点导航控件

在 ASP.NET 1.x 时期, 微软发布了一组 IE Web Control 控件 (非内置控件), 其中包括一个实现树形结构的 TreeView 控件, 它就是典型的站点导航控件。在 ASP.NET 2.0 中, 不仅包含和扩展了原有的 TreeView, 而且还新增了另外两个控件。图 1-2 所示显示了站点导航控件的应用效果。

站点导航控件主要用于实现站点页面导航功能。如图 1-2 所示, 站点导航控件包括 SiteMapPath、Menu 和 TreeView 控件。SiteMapPath 控件能够根据站点导航信息, 准确定位当前页面所处整个 Web 站点的位置, 同时, 使用层次化表示方法, 将位置信息显示为有序的静态文本或者超链接。另外, 还可以通过调整相关属性, 自定义位置信息的外观以及实现数据绑定等功能。Menu 控件可构建与 Windows 应用程序类似的菜单。该控件不仅可与多种数据源控件集成, 而且还可以支持自定义外观、事件处理等功能。TreeView 控件主要用于显示树形结构, 该树与 Windows 资源管理器中的树类似。同时, 该控件还支持数据绑定、自定义外观等功能。以上有关站点导航控件的内容, 将在第 7 章中详细讲解。

2. 数据控件

ASP.NET 2.0 的数据控件可细分为两类: 一类是数据源控件, 另一类是数据绑定控件。数据源控件包括 SqlDataSource、AccessDataSource、XmlDataSource、SiteMapDataSource 和 ObjectDataSource。这些控件主要实现连接不同数据源、数据检索和修改功能, 例如, 查询、排序、分页、筛选、更新、删除和插入等。数据绑定控件主要包括 GridView、DetailsView 和 FormView 等。这些控件可与数据源控件配合, 将获取的数据以不同形式显示在页面上。由于数据源控件和数据绑定控件支持良好的可视化设计时功能, 因此, 当利用 Visual Studio 2005 实现数据访问时, 甚至不需要编写任何代码就能够完成任务。图 1-3 所示显示了在数据源控件支持下, GridView 和 DetailsView 控件的应用效果。

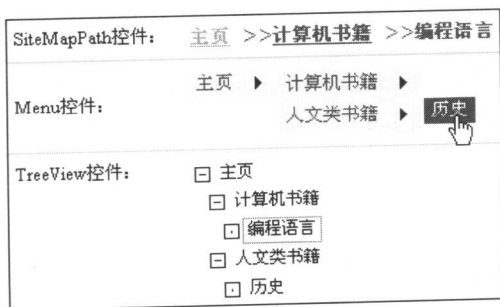


图 1-2 站点导航控件应用效果图

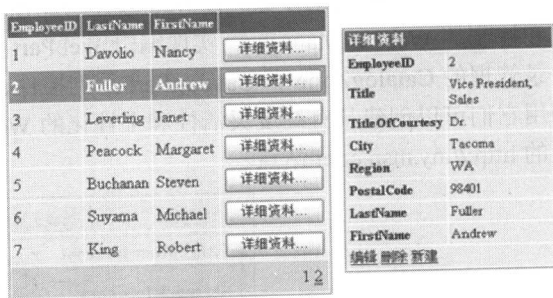


图 1-3 数据控件应用效果图

图 1-3 中包括两个表格。左边的表格列举了数据库表中的数据, 同时, 还提供了分页和显示详细资料的按钮。该表格使用 GridView 控件创建。右边的表格显示左表中被选中行的详细数据, 还提供了编辑、删除和新建等功能按钮。该表格使用 DetailsView 控件创建。有关数据控件以及图 1-3 中所示应用的实现, 将在第 9、10 和 11 章中详细介绍。

3. 登录系列控件

很多 Web 站点都提供了用户登录验证以及相关功能。常见功能包括用户登录、创建新用户等。总体而言, 实现这些功能的基本方法比较类似, 然而, 由于开发人员的水平差异, 可能造成一些

不必要的漏洞和错误。为了解决这个问题, ASP.NET 2.0 将与登录验证相关的实现集成为登录系列控件。这些控件包括: Login、LoginName、LoginView、LoginStatus、PasswordRecovery、ChangePassword 和 CreateUserWizard。利用这些控件可以轻松实现登录验证、创建新用户、显示登录状态、显示登录用户名、更新和重置密码等。出于灵活性考虑, 登录系列控件不仅提供了大量成员对象, 而且还支持自定义模板功能(部分控件支持)。另外, 登录系列控件还加强了安全方面的控制。例如, 默认情况下, 登录系列控件是以纯文本形式工作于 HTTP 上。如果应用程序对安全性十分关注, 那么可以使用带 SSL 加密的 HTTPS。图 1-4 和图 1-5 所示显示了登录系列控件中 Login 控件和 CreateUserWizard 控件的应用效果。

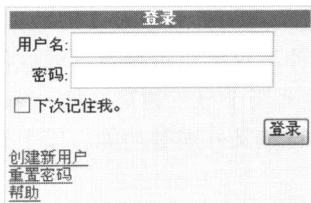


图 1-4 Login 控件应用效果图

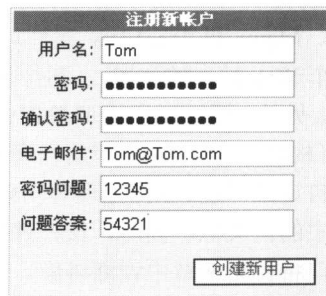


图 1-5 CreateUserWizard 控件应用效果图

Login 控件用于实现用户登录验证, CreateUserWizard 控件用于创建新用户。可能读者会沿袭 ASP.NET 2.0 之前的思路, 认为实现以上两种应用需要添加多个文本框、按钮等元素, 并设计实现数据库、数据访问逻辑等。实际上, 使用 Login 和 CreateUserWizard 控件实现以上应用, 非常简单, 简单到不需要编写任何代码, 不需要自建数据库, 只需添加和设置控件即可。在本书的第 16 章中详细探讨了包括以上两个控件在内的 7 个登录系列控件的应用。

4. Web 部件

Web 部件是 ASP.NET 2.0 新增功能。如果曾经开发应用过 SharePoint, 那么对 Web 部件应该不会感到陌生。ASP.NET 2.0 的 Web 部件功能包括多个服务器控件, 例如, 实现 WebPart 控件管理和控制的 WebPartManager、实现编辑 WebPart 控件的 EditorZone 和 EditorPart 系列控件、实现目录管理的 CatalogZone 和 CatalogPart 系列控件、实现 WebPart 通信的 ConnectionZone 控件等。利用它们可以创建具有高度灵活性和个性化的 Web 站点。图 1-6 所示展示了利用 Web 部件技术构建的 <http://my.msn.com> 站点。

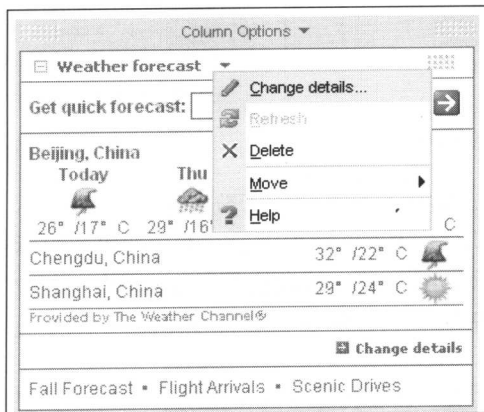


图 1-6 Web 部件应用效果图

使用 Web 部件构建的页面可以为用户提供多种个性化功能。例如,使用图中“Move”选项对 WebPart 控件进行布局调整,使用“Delete”和“Change details...”选项对 WebPart 控件进行配置和管理等。另外,使用 Web 部件还可以实现 WebPart 控件之间的通信、目录管理等功能。有关 Web 部件的详细内容,可参考本书第 14 章。

5. 其他服务器控件

除上文介绍的 4 类控件之外,ASP.NET 2.0 新增其他服务器控件。这些控件包括 BulletedList、HiddenField、FileUpload、ImageMap、MultiView (View) 和 Wizard。在本书的第 8 章中详细介绍了这些控件的基本功能、应用方法等。

1.3.2 母版页

每个站点都应具有统一的风格和布局。不同技术时代,给出了不同的解决方案。例如,在 ASP 技术时期,使用#include 命令,在 ASP.NET 1.x 时期,主要使用用户控件。ASP.NET 2.0 技术时期,则可以使用构建页面布局框架的技术特性——母版页。

母版页是扩展名为.master 的文件,其代码内容和结构与普通.aspx 文件类似。注意代码中包括一个或多个 ContentPlaceHolder 控件。在创建母版页时,需要将页面公共部分存储于母版页中,例如,页面公用的页头、页尾等,而页面非公共部分则使用 ContentPlaceHolder 控件实现占位。内容页文件虽然扩展名为.aspx,但是代码内容和结构与普通.aspx 文件代码相距甚远,其代码分为两个部分:代码头声明一个或多个 Content 控件。开发人员需要在内容页代码头绑定母版页,同时,将页面非公共部分内容设置在 Content 控件标签之间。在运行时,用户不能直接请求母版页,只能请求访问内容页。此时,母版页和内容页将合并生成结果页,结果页面包含页面公共部分和非公共部分的运行结果。图 1-7 所示显示了母版页功能的应用示例图。

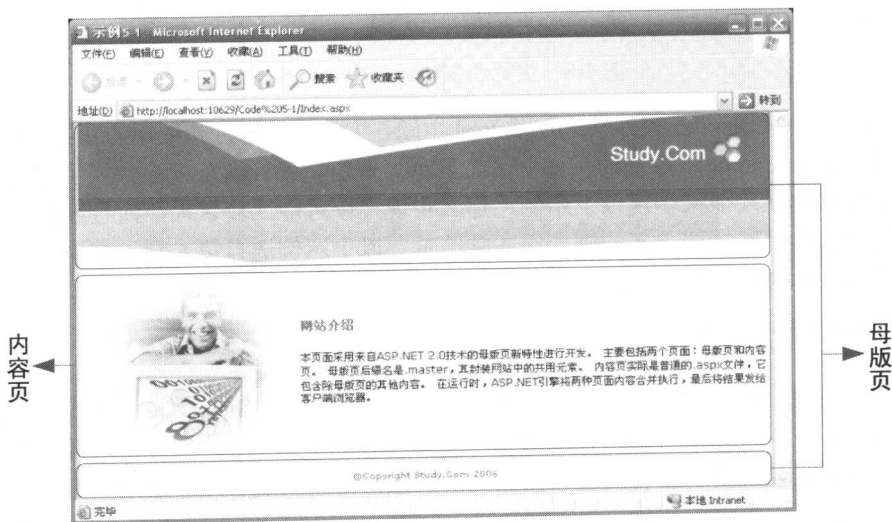


图 1-7 母版页功能应用效果图

图 1-7 所示的内容节选自第 5 章的示例。在该应用程序中,母版页实现了页头和页尾等公共部分,中间的非公共部分则包含在内容页中。这样,母版页可被绑定在不同的内容页中,从而提高了开发人员工作效率,降低了开发和维护强度。

以上概括性介绍了母版页功能。在第 5 章中,读者不仅能够更加详细地了解母版页的基本功能,而且还可以接触到一些高级应用。例如,在母版页中实现嵌套,即父母版页嵌套一个子母版页;内容页绑定子母版页。又如,在内容页中访问母版页控件和属性、动态加载母版页、母版页