

ASP.NET 循序渐进 入门篇

ASP.NET 2.0

程序开发详解

杨云 王毅 编著

微软公司高级软件工程师精心编写

内容全面，从ASP.NET 2.0的运行原理到最新的ASP.NET AJAX技术丰富的技术案例，全面、高效掌握ASP.NET 2.0开发技术
使用C# 2.0描述、以Visual Studio 2005作为开发平台进行讲解。

 人民邮电出版社
POSTS & TELECOM PRESS



ASP.NET 循序渐进 入门篇

ASP.NET 2.0

程序开发详解

杨云 王毅 编著

人民邮电出版社
北京

图书在版编目 (CIP) 数据

ASP.NET 2.0 程序开发详解 / 杨云, 王毅编著. —北京: 人民邮电出版社, 2007.5

ISBN 978-7-115-15934-2

I. A... II. ①杨... ②王... III. 主页制作—程序设计 IV. TP393.092

中国版本图书馆 CIP 数据核字 (2007) 第 030240 号

内 容 简 介

本书全面系统地讲解了 ASP.NET 2.0 的各个方面的内容, 共分 17 章。全书始终采用知识讲述+代码示例的方式, 读者可以马上通过代码体验到 ASP.NET 的强大。本书涉及 ASP.NET 2.0 的各个方面的内容, 力求让读者迅速掌握 ASP.NET 2.0 程序的设计方式, 能够应用 ASP.NET 完成编程任务。

本书后两章对微软提供的大型示例进行了讲解, 采用数据库构架讲解、系统构架剖析和关键代码讲解的形式来让读者对示例从结构上和功能上进行认识, 然后通过对示例添加一些功能, 让读者学习到如何在现有大型示例的基础上建设自己的应用。

本书讲解细致, 循序渐进, 既适合没有 ASP.NET 基础的初学者直接学习, 也适合 ASP.NET 1.1 的开发人员阅读。书中例子涵盖不同层面, 相信可以让读者全面学习到使用 ASP.NET 技术开发时应该掌握的知识。

ASP.NET 2.0 程序开发详解

- ◆ 编 著 杨 云 王 毅
责任编辑 刘 浩
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 32
字数: 782 千字 2007 年 5 月第 1 版
印数: 1—5 000 册 2007 年 5 月北京第 1 次印刷

ISBN 978-7-115-15934-2/TP

定价: 55.00 元 (附光盘)

读者服务热线: (010)67132692 印装质量热线: (010)67129223



杨云

Microsoft MVP，参加过多项基于微软.NET技术的大型项目，包括企业级应用系统、电信应用系统、政府办公系统等。开发的系统主要分两类，企业级应用包括MIS、CRM、BOSS、集成SPS、VSS扩展开发等，政府级应用包括OA WorkFlow、OfficeSystem扩展开发。

在Microsoft ASP.NET技术体系方面主要的研究方向是：ASP.NET技术在实际项目中的简便开发，典型设计模式在Web开发中的运用。



王毅

微软亚洲工程院(Microsoft ATC)高级软件工程师，长期从事微软.NET产品的研发等工作。

在Microsoft.NET技术体系方面主要的研究方向是：服务器组件的开发，大型站点前台构架，后台数据库的设计，Office插件系统，VSTS的集成和扩展以及Reporting Service的自定义开发等。

序

技术是为市场服务的，当今的互联网络局已经发生了巨大的改变。网络服务的崛起和高速发展已经说明了其市场需求和客户意愿。然而网络服务需要以技术作为驱动力，Web 开发技术作为其重要的实现部分正在走向新的阶段。

微软为了适应新的变革，在 2005 年底成功推出 ASP.NET 2.0 技术。作为对技术延续和改进的重要体现，ASP.NET 2.0 技术进行了重大改进。在中国，ASP.NET 2.0 技术正处于学习和应用的普及期，作为想学习该技术的 IT 从业人员，需要一本系统介绍 ASP.NET 2.0 技术并指导实践的良师益友，这本书应该是一个不错的选择。

本书主题紧扣 ASP.NET 2.0 技术的每个环节，通过理论加实践的方式使读者熟悉知识点的定义并掌握如何使用它们。

本书透彻地讲解了 ASP.NET 2.0 的内部机理、特有特性、ASP.NET 控件。作者为了使读者对知识点有更深刻、系统的理解，在书中第 16 和 17 章介绍了设计规范的两个完整系统。通过这些内容，读者能够对 ASP.NET 2.0 技术有更深入的理解。

本书在 ASP.NET 2.0 技术发布的 11 个月完成，足以说明本书在内容的选择和资料收集上都是按照目前最新成果创作的，是可以信任的。书中吸取了以往 ASP.NET 2.0 技术图书的优点，改进了它们的不足，在很多问题的阐述上有创新或独到之处。

本书由微软中国区 .NET 技术微软最有价值专家（Most Valuable Professionals, MVP）杨云和目前就职于微软亚洲工程院的王毅先生共同合著完成。

杨云先生作为微软 MVP，有过多年的微软技术项目开发和管理经验，他出版过多部关于 .Net/ASP.NET 等方面的专著和文章，在参与微软组织的各种活动中也收集了很多技术反馈和用户体验，这也为本书的创作打下了良好的基础。

王毅先生作为微软亚洲工程院的员工，承担了很多微软对外和对内的开发任务，积累了丰富的 ASP.NET 2.0 使用经验。在本书的创作中，王毅先生和杨云先生通过良好的沟通机制为本书的顺利创作提供了坚实的保障。

我相信这本书能够给您带来对 ASP.NET 2.0 技术的系统认识并将它运用到实际的工作当中。本书在创作中力争照顾到入门的技术人员和有 ASP.NET 1.X 开发经验的人员。我相信在阅读完本书之后，您会发现这是一本好书！



柯淑芬

亚洲区总经理

微软最有价值专家暨社群

前 言

Web 开发技术已经经历了十多个年头, 不管技术如何更新换代都是为了满足市场的需要。在当前的开发领域, 仍然采用服务器端编程模型进行实际的 Web 开发, 事实证明这才是真正能为用户提供丰富体验的技术。

在当前软件需求日趋复杂的大趋势下, 开发人员应该把更多精力投入到改善设计和完善用户体验上。ASP.NET 开发技术作为微软主力开发技术经历了 1.0、1.1 的历史阶段, 在不断总结和听取反馈后于 2005 年底正式发布了 ASP.NET 2.0。

ASP.NET 2.0 延续了 ASP.NET 1.1 的编程模式、代码设计、实现方法、语法模型。其变化最大之处就是 ASP.NET 2.0 简化了开发的过程, 给予设计人员更多思考软件设计的时间。

与 ASP.NET 1.1 相比, ASP.NET 2.0 开发的项目将减少 40%~50% 的人工编码量, 更多的代码将由 IDE 自动生成。ASP.NET 2.0 新增加的数十个控件基本涵盖了开发人员以前经常需要手动开发的功能, 如用户登录、用户创建、用户管理、WebPart 和数据源控件等。

在编译和部署方面也为开发人员提供了更多的选择, 不管是采用普通编译还是预编译都将比原来更加安全和更有效率。

ASP.NET 2.0 包含网站管理工具, 使网站管理人员可以使用基于 Web 的界面管理站点。ASP.NET 2.0 还增强了缓存管理等功能, 总体来说 ASP.NET 2.0 是具有里程碑意义的。

本书主要面向使用 C# 2.0 语言开发 ASP.NET 2.0 应用的开发人员, 共 17 章, 主要偏重于基础和覆盖面广的知识点, 从 ASP.NET 2.0 的运行原理、控件使用、C#2.0 基础知识到时尚的 AJAX 和 ASP.NET AJAX 技术, 力争使读者通过学习能掌握如何使用 Visual Studio 2005 开发基于 ASP.NET 2.0 的应用。

本书对于知识点都采取引领的方法, 使读者能够一步步理解知识点, 增加学习的兴趣。对于比较重要的理论知识点都安排有相应的短小实例代码进行讲解。读者不仅可以按照书中的示范编写代码来巩固知识点, 还可以通过附赠的光盘获取现成的源代码。

除了配合 ASP.NET 特性讲解的小示例, 在本书第 16 章和 17 章还讲解了两个流行的微软专门为 ASP.NET 2.0 用户提供的大型站点示例。Time Tracker 系统会使读者学习到如何开发使用 ASP.NET 2.0 的技术搭建一个工作任务追踪站点, PetShop 4 则更详细示例了如何综合利用 ASP.NET 2.0 各种优秀特性构建一个比较大型的在线购物的站点。这些例子充分显示了 ASP.NET 2.0 在方便开发人员, 减少代码量方面做出的努力。

致谢

感谢微软公司亚洲区总经理柯淑芬女士为本书写序, 感谢母亲对我无微不至的照顾和支持, 感谢本书合著者王毅为本书付出的努力以及其家人对他的支持和理解。

杨 云

2007 年 3 月

光盘使用说明

结合光盘中提供的实例阅读本书是作者推荐的阅读方式，在安装项目代码前请仔细阅读本说明。

本光盘包括书中所有实例的工程文件和源代码。本光盘所提供的系统源代码都经过编译通过并测试其功能（功能 BUG 不在所说范围）。本光盘代码读者可以自行修改和编译出自己的版本。

本书的实例工程可以通过 VS.NET 2005 直接加载，无需 IIS 服务器即可打开 Web 站点。

为方便读者使用，每个项目都提供项目所有的源代码和完整的配置信息，可以直接运行站点。

对于系统中使用的非作者开发的控件，保留其原始公司或作者的一切权利。

光盘的目录与书的章节相对应，如光盘中“第 3 章”文件夹对应书的第 3 章，依此类推。

目 录

第 1 章 .NET 框架概述	1
1.1 .NET Framework 体系架构	1
1.2 撼动未来的 .NET Framework 2.0	3
第 2 章 .NET 编程的基础知识	6
2.1 C# 编程基础	6
2.1.1 C# 的变量系统	6
2.1.2 常量	8
2.1.3 预定义数据类型	8
2.1.4 流程控制	12
2.1.5 面向对象的特性	17
2.1.6 运算符	25
2.1.7 事件和委派 (delegate)	28
2.2 C# 2.0 新特性	31
2.2.1 范型编程	32
2.2.2 范型委派	39
2.2.3 匿名方法	40
2.2.4 部分类 (partial class)	40
2.3 ADO .NET 基础	41
2.3.1 Provider (提供者程序)	41
2.3.2 使用 Connection	42
2.3.3 使用 Command	43
2.3.4 使用 DataReader、DataSet 和 DataAdapter	47
2.4 XML Web Service 基础	49
2.4.1 XML Web Service 基础	49
2.4.2 XML Web Service 的体系结构	49
2.4.3 XML Web Service 的消息协议	50
第 3 章 Visual Studio.NET 2005 开发环境	52
3.1 安装 VS2005	52
3.2 创建和打开 Web 站点	52
3.3 使用内置的 ASP.NET Deployment Server	54
3.4 迁移现有的 VS2002/VS2003 Web 站点	55

3.5	编辑 Web 站点	58
3.6	使用服务器控件	60
3.7	创建事件处理程序	62
3.8	验证 HTML 源码的可用性	62
3.9	使用 Visual Studio 的 Intellisense	64
3.9.1	列出对象成员	65
3.9.2	显示方法参数信息	65
3.9.3	快速信息	65
3.9.4	自动完成	66
3.9.5	C#相关的智能感知	66
3.10	对重构的支持	68
3.11	调试和测试	71
3.12	页面与代码的组织	72
3.13	ASP.NET 2.0 应用程序文件夹	75
3.14	ASP.NET 2.0 的预编译	79
第 4 章	ASP.NET 2.0 体系结构	83
4.1	代码模型	83
4.2	代码的结构	84
4.3	编译模型	85
4.4	扩展性与管道技术	86
4.5	缓存技术	87
第 5 章	ASP.NET 2.0 网络服务	90
5.1	网络服务 (Web Service) 基础	90
5.1.1	Web Service 的概念	90
5.1.2	Web Service 的基础技术	90
5.1.3	Web Service 的软件支持	91
5.1.4	Web Service 的编码模型	91
5.1.5	使用 Visual Studio 2005 开发 Web Service	92
5.2	Web Service 的演进方向	95
5.3	基于接口的服务约定	95
5.4	更多的 XSD/WSDL 改进	97
5.5	更好的互操作性	98
5.6	为 Windows Communication Foundation (WCF) 做好准备	100
第 6 章	ASP.NET 2.0 功能增强控件	101
6.1	数据源控件	101
6.1.1	SqlDataSource 数据源控件	102

6.1.2	XmlDataSource 数据源控件	105
6.1.3	ObjectDataSource 数据源控件	108
6.1.4	AccessDataSource 数据源控件	108
6.1.5	SiteMapDataSource 数据源控件	109
6.2	GridView 控件	109
6.2.1	使用 GridView 显示数据	110
6.2.2	使用自定义数据列	114
6.2.3	使用模板列	117
6.2.4	删除数据	120
6.2.5	控件参数	125
6.2.6	利用数据源控件缓存数据	126
6.3	DetailsView 控件	127
6.3.1	使用 DetailsView 显示、编辑和删除数据	127
6.3.2	插入新记录	130
6.3.3	使用模板	131
6.3.4	同时使用 GridView 和 DetailsView	134
6.4	TreeView 控件	135
6.4.1	使用静态数据	136
6.4.2	使用动态数据	137
6.4.3	通过数据库填充控件	137
6.5	Login 控件	140
6.6	PasswordReovery 控件	141
6.7	LoginStatus 和 LoginName 控件	142
6.7.1	LoginStatus 控件	142
6.7.2	LoginName 控件	143
6.8	LoginView 控件	143
6.9	CreateUserWizard 控件	144
6.10	BulletedList 控件	146
6.11	ImageMap 控件	148
6.12	MultiView 和 View 控件	150
6.13	Wizard 控件	152
6.14	Panel 控件	155
6.15	FileUpload 控件	158
6.16	HiddenField 控件	159
6.17	Substitution 控件	160
6.18	TextBox 控件	162

第 7 章 ASP.NET 2.0 中的 MasterPage

7.1	新建 MasterPage	163
-----	---------------------	-----

7.2	在内容页嵌入 MasterPage	164
7.3	使用多个内容区域和默认内容	166
7.4	动态使用 MasterPage	169
7.5	在运行时访问 MasterPage	172
7.6	嵌套的 MasterPage	174
第 8 章	ASP.NET 2.0 成员和角色管理	177
8.1	认证和授权	177
8.1.1	IIS 和 ASP.NET 用户认证流程	177
8.1.2	认证 (Authentication)	178
8.1.3	授权 (Authorization)	178
8.2	ASP.NET 2.0 用户认证	178
8.2.1	使用 ASP.NET 管理工具添加用户	181
8.2.2	使用 CreateUserWizard 创建用户	184
8.2.3	改变默认的 Provider 设置	184
8.2.4	个性化 CreateUserWizard 控件	185
8.2.5	使用 Login (登录) 相关的控件	187
8.3	ASP.NET 角色管理系统	191
8.3.1	角色管理	192
8.3.2	角色管理和成员管理的关系	192
8.3.3	应用角色管理	192
8.3.4	修改 <RoleManager> 节点	194
8.3.5	使用用户角色控件	195
8.4	使用 Membership/Role API	197
8.4.1	使用 Membership API 管理用户	197
8.4.2	使用 Role API 进行用户角色管理	200
8.5	ASP.NET 的 MemberShip Provider	204
8.5.1	SqlMembershipProvider	204
8.5.2	ActiveDirectoryMembershipProvider	206
8.6	实现自定义的 MembershipProvider	207
8.7	基于角色的站点导航	212
第 9 章	窗体页设计技巧	216
9.1	Page 类的新事件	216
9.2	添加标题	218
9.3	设置焦点	218
9.4	为 Form 设定默认按钮	219
9.5	更好的输入验证控件	220
9.6	使用 Page.Items 字典	222

9.7	使用跨页面传送功能	223
9.8	高速缓存和 SQL Server Invalidation 功能	226
9.9	配置 SQL Server Invalidation	226
9.10	使用 SQL Server Invalidation 和数据源控件	227
9.11	通过编程方式使用 SQL Server Invalidation	228
9.12	高速缓存的其他改进	229
9.13	使用页面高速缓存	229
第 10 章	使用 ASP.NET 2.0 Web Part 框架	231
10.1	常用 WebPart 控件	231
10.1.1	WebPartManager 控件	231
10.1.2	WebPartZone 控件	233
10.1.3	CatalogZone 控件和所属 CatalogPart 控件	237
10.1.4	EditorZone 和所属 EditorPart 控件	242
10.1.5	ConnectionZone 控件和信息通信	245
10.2	个性化 WebPart 的数据存储和转移	247
第 11 章	创建 ASP.NET 服务器控件	250
11.1	ASP.NET 服务器控件概述	250
11.2	服务器控件项目的设置	254
11.3	服务器控件的呈现	255
11.3.1	输出控件的内容	256
11.3.2	为 HTML 元素添加属性	257
11.3.3	控件的适应性	258
11.4	开始创建服务器控件	260
11.5	创建复合控件	268
11.6	为控件添加更多功能	279
11.6.1	为控件添加输入验证	279
11.6.2	控件的子属性	281
11.6.3	为 Register 控件增加嵌套子属性	284
11.7	控件的回调示例	286
11.7.1	异步请求	286
第 12 章	ASP.NET 2.0 中的页面主题/皮肤	290
12.1	页面主题概述	290
12.2	页面主题的运用	291
12.2.1	App_themes 目录	291
12.2.2	全局页面主题和局部页面主题	292
12.3	皮肤文件和主题的使用	294

12.4	使用样式表主题	304
12.5	资源与主题	307
12.6	动态加载页面主题	309
第 13 章 ASP.NET 2.0 配置详解		314
13.1	ASP.NET 配置的基本结构	314
13.1.1	.NET 应用程序的配置体系	314
13.1.2	ASP.NET 配置结构	314
13.1.3	.NET 配置文件基本结构	315
13.1.4	配置区域和配置组	315
13.1.5	添加自定义的配置节	317
13.1.6	使用 location 节点和 path 属性	318
13.1.7	ASP.NET 常用配置节点	319
13.2	获取配置信息	323
13.3	使用 ASP.NET 配置管理接口	326
13.3.1	使用配置管理接口访问程序配置	326
13.3.2	对配置内容加密	327
13.4	使用 ASP.NET 配置工具	329
13.4.1	使用 ASP.NET 管理控制台	329
13.4.2	使用 ASP.NET 管理站点	331
13.4.3	使用 ASPNET_REGSQL 工具	332
13.4.4	使用 ASPNET_REGIIS 工具	333
13.5	ASP.NET 页面配置	333
13.6	配置 ASP.NET 进程模型	335
第 14 章 让站点符合国际化要求		337
14.1	国际化和本地化	337
14.1.1	国际化和本地化	337
14.1.2	ASP.NET 2.0 对国际化的支持	338
14.2	自动检测浏览器语言	338
14.2.1	在浏览器中设置语言偏好	338
14.2.2	使 ASP.NET 页面能够自动检测浏览器语言文化设定	338
14.3	ASP.NET 程序中的本地化	341
14.3.1	无代码本地化	342
14.3.2	从代码中访问资源文件	347
第 15 章 AJAX 和 ASP.NET AJAX		348
15.1	AJAX 简介	348
15.1.1	Web 技术的进化	348

15.1.2	Web 应用程序存在的问题	350
15.1.3	AJAX	351
15.1.4	为什么要使用 AJAX	351
15.1.5	AJAX 有待解决的问题	352
15.2	AJAX 的核心技术	352
15.2.1	JavaScript 简介	353
15.2.2	文档对象模型 (DOM)	354
15.2.3	CSS (层叠样式表)	357
15.2.4	XMLHttpRequest 对象	359
15.2.5	AJAX 应用示例	363
15.3	ASP.NET AJAX Extension 简介	371
15.3.1	ASP.NET AJAX 框架的由来	371
15.3.2	ASP.NET AJAX 框架的组成	372
15.3.3	ASP.NET AJAX 的获取和安装	374
15.3.4	ASP.NET AJAX 的客户端脚本库	374
15.3.5	创建 ASP.NET AJAX 网站	374
15.3.6	ASP.NET AJAX 版本的 Hello World	377
15.3.7	UpdatePanel 和局部重绘	379
15.3.8	AJAX 风格的留言板	381
第 16 章	Time Tracker Starter Kit 解析	388
16.1	Time Tracker 的安装和配置	388
16.1.1	下载和安装	388
16.1.2	创建一个新的 Time Tracker 站点	390
16.2	Time Tracker 的系统设计	391
16.2.1	Time Tracker 的功能模块	391
16.2.2	功能模块的划分	392
16.2.3	Time Tracker 的数据库结构	394
16.2.4	数据库的表设计	396
16.3	使用 Time Tracker 站点工作包	400
16.3.1	运行 Time Tracker 站点	400
16.3.2	创建用户	401
16.3.3	创建新项目	402
16.3.4	Time Tracker 站点模块的设计	403
16.4	扩展 Time Tracker	409
16.4.1	建立更多的角色	410
16.4.2	修改登录方式	411
16.4.3	为页面增加功能	412
16.4.4	全球化/本地化页面	417

第 17 章 PetShop 4 宠物商店精讲	421
17.1 PetShop 4 的架构设计	421
17.1.1 PetShop 4 数据库设计	424
17.1.2 数据访问层设计思想	429
17.1.3 业务逻辑层设计思想	445
17.1.4 同步异步模式设计思想	452
17.1.5 配置文件模块设计思想	459
17.1.6 缓存模块设计思想	470
17.2 PetShop 4 交互 UI 层设计	476
17.2.1 UI 层总体设计	476
17.2.2 用户自定义控件设计	479
17.2.3 用户交互页设计	487

第 1 章 .NET 框架概述

本书讨论的 .NET Framework (框架) 是一个开发平台, 也是实施微软 .NET 策略的一项重要举措, 它引入了很多新的概念、技术和术语。

1.1 .NET Framework 体系架构

.NET 框架提供了必要的编译时和运行时技术来产生和运行 .NET 应用。 .NET 框架必须运行在操作系统之上。目前, .NET 框架可以运行在 Win 32 操作系统上。另外, .NET Compact Framework (精简框架) 能够运行在 Windows CE 和其他嵌入式操作系统上。

.NET 框架的体系架构如图 1-1 所示, 在此对它的 3 个重要组成部分进行解释, 并引入几个常用的术语。

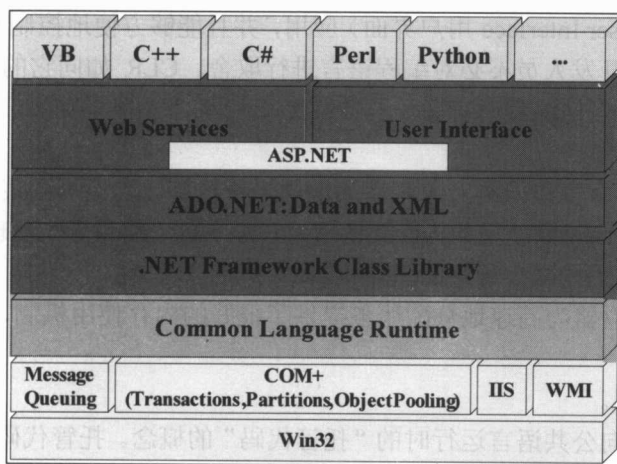


图 1-1 .NET Framework 体系架构图

(1) CLR (Common Language Runtime, 公共语言运行时)。

公共语言运行时简化了应用的开发, 提供了健壮安全的执行环境, 支持多种编程语言, 并且简化了应用的发布和管理。它提供了很多的服务, 包括:

- 元数据管理;
- MSIL (Microsoft Intermediate Language 微软中间语言) 到本机代码的转换;
- 代码管理 (加载和执行);
- 自动进行对象生命期的管理;
- 强制代码访问安全;
- 调试应用和跟踪代码执行;

- 类型安全验证;
- 执行管理, 提供结构化的例外处理;
- 多线程支持;
- 托管代码、COM 对象和现有 DLL (非托管代码和数据) 之间的互操作;
- 集成代码, 支持 .NET 框架类库;

(2) The .NET Framework Class Library (.NET 框架类库)。

.NET 框架类库展现了 CLR 的特性, 提供给每个开发者都需要的高层次的服务。.NET 框架类库中有成百上千个类, 分组在不同的 Namespace (命名空间) 中。

(3) ASP.NET。

ASP.NET 是 .NET 框架的核心组成部分, 也是本书的核心。ASP.NET 是 CLR 中的可编程框架, 能够在服务器上创建强大的 Web 应用。ASP.NET Web Forms 提供了简单有效的方法来动态建立 Web 用户接口网页。ASP.NET Web Services 使用生成块来结构化地分配基于 Web 的应用。Web Services 基于开放的 Internet 标准, 如 HTTP 和 XML。

整个 .NET 框架有着严谨合理的体系架构, 具有以下主要特征。

(1) 与编程语言无关。

对于开发人员来讲, 在选择编程语言的时候往往处于两难的境地。例如, 开发人员使用 C/C++ 能够控制底层系统, 能够管理内存、创建进程, 等等。开发人员使用 Visual Basic 6 能够快速地生成 UI (User Interface 用户界面) 应用, 并且能够方便地控制 COM 对象和数据库。而在 .NET 框架中, 开发人员不必对编程语言进行取舍。CLR 如同它的名字——公共语言运行时那样, 能够支持任何编程语言。

(2) 垃圾回收机制。

垃圾回收是使计算机能检测对象生命周期的一种机制。当检测到某个对象不再需要使用时, 它能够自动释放该对象使用的内存 (调用用户编写的“结束者”清理例程)。一些垃圾回收器会压缩内存, 并因此减少程序的工作集。

.NET 的垃圾回收器为高速地分配服务提供了很好的内存使用机制, 并且不会带来长期碎片的问题。

(3) 互操作性强。

.NET 引入了面向公共语言运行时的“托管代码”的概念。托管代码具有许多优势, 其中包括自动内存管理、属性化编程以及一个公共类型系统。但是, 托管代码强大的功能使其从根本上不同于非托管代码 (如 Windows API 和 COM 对象)。使托管对象和非托管对象协同工作的过程称为互操作性。

.NET 中的互操作性涉及 3 个不同的方面:

- 在托管代码中使用 COM 对象, 即将现有的 COM 对象用于托管代码;
- 使用 .NET 创建 COM 对象, 即使用 .NET 便捷的开发生性能来开发传统的 COM 对象;
- 从托管代码中使用 Windows API, 即 .NET 框架为许多 Windows API 提供了托管代码替代, 但有时可能需要调用动态链接库 (dll) 中的函数。

(4) 安全性。

.NET 框架提供了一个强健的安全系统, 该系统能够限制代码在严格约束的、管理员