

C#和ASP.NET

程序设计教程

木林森 编著



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



C# 和 ASP.NET 程序 设计教程

木林森 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书从实用角度出发,对 C# 编程及怎样用 ASP.NET 进行 Web 应用程序的开发做了全面系统的介绍。全书共分 8 章。第 1 章对什么是 .NET 及 .NET 体系结构和框架做了介绍;第 2 章介绍了 C# 编程的基础知识;第 3 章介绍了怎样用 C# 进行面向对象编程;第 4 章对接口、名字空间、文件管理及注册表编程做了介绍;第 5 章对 Web 动态开发技术(如 CGI、服务器 API、PHP 和 ASP)做了介绍;第 6 章介绍了用 ASP.NET 开发 Web 应用程序的基础知识;第 7 章介绍了怎样用 ADO.NET 进行 Web 数据库编程;第 8 章通过一个综合实例(在线论坛)对前面所学的知识做了贯穿。

本书对 C# 和 ASP.NET 中的每个知识点、概念和难点,都力求以较精练的语言进行讲解并配以必要的实例,读者只要照书中的实例上机操作,便可全面掌握 C# 和 ASP.NET 编程的思路和开发技巧与体系。

本书适合 C# 和 ASP.NET 开发人员使用,并可作为各类培训班的教材。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: C# 和 ASP.NET 程序设计教程

作 者: 木林森

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮政编码:100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 章忆文

印 刷 者: 北京密云胶印厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 **印张:** 21.5 **字数:** 522 千字

版 次: 2002 年 1 月第 1 版 2002 年 1 月第 1 次印刷

书 号: ISBN 7-900637-30-3

印 数: 0001~5000

定 价: 38.00 元

前 言

.NET 是微软公司推出的新一代互联网软件和服务战略,它是一种面向网络、支持各种用户终端的开发平台环境。.NET 使微软公司现有的软件在网络时代不仅适用于传统的 PC 机(个人电脑),而且能够满足呈现强劲增长势头的新设备(如手机和 PDA 等)的需要。微软的宏伟目标是让 .NET 彻底改变软件的开发方式、发行方式、使用方式,使软件变成服务。.NET 的核心就是要搭建新一代 Internet 平台,用于解决网站之间的协同合作问题,从而最大限度地获取信息。在 .NET 平台上,不同网站之间通过相关的协定联系在一起,网站之间形成自动交流,协同工作,提供最全面的服务。

C# 是微软公司专门为 .NET 量身订做的编程语言,它与 .NET 有着密不可分的关系。例如,C# 的类型其实就是 .NET 框架所提供的类型,C# 本身并无类库,而是直接使用 .NET 框架所提供的类库。另外,类型安全检查、自动资源回收、结构化异常处理等也都是交由 CLR (公共语言运行时)处理的。因此,C# 是最适合开发 .NET 应用的编程语言。

ASP.NET 是微软公司继 ASP(活动服务器页面)之后推出的用于创建、管理和部署 Web 应用程序的新的理想平台。它是使用 .NET 框架提供的编程类库构建而成的。使用 ASP.NET,创建、管理和部署 Web 应用程序变得非常容易。

本书从实用角度出发,对 C# 编程及怎样用 ASP.NET 进行 Web 应用程序的开发做了全面系统的介绍。在写作本书时,我们不是简单地介绍 C# 和 ASP.NET 的使用方法,而是通过范例来步步引导读者快速掌握用 C# 和 ASP.NET 进行编程的思路、技巧与体系。读者在阅读本书时,不应该仅仅满足于书中所介绍的具体内容,而应当掌握用 C# 和 ASP.NET 进行编程的基本方法,并在此基础上举一反三,通过自己的实践去扩展所学的知识并切实应用到实际开发工作中去。

本书所有范例的源程序都放在随书光盘中,使用时只需将这些源程序拷贝到硬盘中即可。另外,为了方便读者学习本书,我们在随书光盘中提供了 .NET Framework SDK。

本书由木林森负责编写,高峰霞、奚红宇对全书做了仔细审阅。参与本书部分编写工作的还有高越明、高星泉、罗丽琼、王世航、高月华、邵光伟、叶伟东、张军、王强、林小莉、王志军、杨晏文、冯建军和孙智伟。本书的录排工作由刘珊负责。

本书在编写过程中得到了清华大学出版社的大力支持,在此表示衷心的感谢。由于笔者学识所限,缺点和错误在所难免,竭诚欢迎广大读者批评指正。

作 者

(liaobs@263.net)

2001 年 10 月

目 录

第 1 章 .NET 初步	1
1.1 什么是 .NET	1
1.2 Internet 与 Microsoft 平台的演进	1
1.3 .NET 体系结构	3
1.4 .NET 框架	5
1.4.1 公共语言运行时	5
1.4.2 服务框架	8
1.5 ASP.NET	8
1.6 思考与练习	9
第 2 章 C# 编程基础	10
2.1 概述	10
2.2 编写简单的 C# 程序	10
2.2.1 编译程序 ch2_1.cs	11
2.2.2 名字空间	12
2.2.3 程序的输入和输出	12
2.2.4 命令行参数	13
2.2.5 注释	14
2.3 变量和常量	14
2.3.1 变量	14
2.3.2 常量	15
2.4 数据类型	15
2.4.1 简单类型	15
2.4.2 结构类型	17
2.4.3 枚举类型	18
2.4.4 类型	19
2.4.5 数组类型	19
2.4.6 代表元类型	21
2.5 类型转换	23
2.5.1 隐式转换	23
2.5.2 显式转换	24
2.5.3 Boxing 和 Unboxing	27

2.6	操作符	28
2.6.1	算术操作符	28
2.6.2	赋值操作符	29
2.6.3	比较操作符	29
2.6.4	逻辑操作符	29
2.6.5	位操作符	30
2.6.6	is 操作符	31
2.6.7	三元操作符	31
2.6.8	自增和自减操作符	32
2.6.9	as 操作符	33
2.6.10	new 操作符	34
2.6.11	typeof 操作符	34
2.6.12	sizeof 操作符	35
2.6.13	checked 和 unchecked 操作符	35
2.6.14	操作符优先级	37
2.7	控制语句	37
2.7.1	选择语句	37
2.7.2	循环语句	40
2.8	预处理指令	43
2.8.1	#define 和 #undef	43
2.8.2	条件编译指令	44
2.8.3	#error 和 #warning	45
2.8.4	#line	45
2.9	异常处理	46
2.9.1	try-catch 结构	46
2.9.2	try-finally 结构	48
2.9.3	try-catch-finally 结构	49
2.10	编译选项	49
2.11	思考与练习	52
第3章	面向对象编程	53
3.1	概述	53
3.2	类	53
3.2.1	类的声明	53
3.2.2	类的成员	54
3.2.3	this 关键字	55
3.2.4	静态成员和非静态成员	56
3.2.5	成员常量	57

3.2.6	构造函数和析构函数	58
3.3	方法	60
3.3.1	方法的声明	60
3.3.2	值参数和引用参数	61
3.3.3	输出参数	61
3.3.4	数组参数	62
3.3.5	静态方法与非静态方法	63
3.3.6	方法重载	64
3.3.7	操作符重载	65
3.4	域和属性	68
3.4.1	域	68
3.4.2	属性	70
3.5	事件	73
3.6	索引指示器	78
3.7	继承	80
3.7.1	简介	80
3.7.2	base 关键字	81
3.7.3	隐藏	82
3.7.4	多态性	83
3.8	抽象类	90
3.9	密封类	93
3.10	思考与练习	93
第4章	深入 C# 编程	95
4.1	接口	95
4.1.1	接口定义	95
4.1.2	接口继承	96
4.1.3	访问接口成员	96
4.1.4	接口实现	98
4.1.5	显式接口成员	101
4.2	名字空间	103
4.2.1	名字空间的声明	103
4.2.2	简单实例	104
4.2.3	使用指示符	106
4.3	文件管理和操作	107
4.3.1	System. IO 名字空间	107
4.3.2	FileSystemInfo 类	108
4.3.3	Directory 和 DirectoryInfo 类	109

4.3.4	File 和 FileInfo 类	113
4.3.5	按文本模式读写	118
4.3.6	按二进制模式读写	119
4.4	注册表编程	121
4.4.1	Registry 和 RegistryKey 类	121
4.4.2	一个简单实例	123
4.5	思考与练习	125
第 5 章	Web 开发技术回顾	126
5.1	概述	126
5.2	CGI 技术	126
5.3	服务器 API	127
5.4	PHP	128
5.5	ASP 编程基础	129
5.5.1	ASP 的工作流程	129
5.5.2	一个简单的 ASP 程序	130
5.5.3	使用 ASP 命令	132
5.5.4	多脚本语言混合编程	133
5.5.5	包含文件	134
5.5.6	ASP 内建对象	135
5.5.7	Request 对象	135
5.5.8	Response 对象	143
5.5.9	使用 Cookie	148
5.5.10	Application 对象	150
5.5.11	Session 对象	151
5.5.12	Global.asa 文件	154
5.5.13	Server 对象	155
5.5.14	ASP 组件	158
5.6	ASP.NET 与 ASP 的比较	166
5.7	思考与练习	166
第 6 章	ASP.NET 编程基础	168
6.1	Web Form	168
6.1.1	一个简单实例	168
6.1.2	页面处理过程	169
6.1.3	页面状态	170
6.1.4	页面事件	171
6.1.5	页指示符	172

6.2	HTML Server Controls	172
6.2.1	HTML Server Controls 的层次结构	172
6.2.2	HTML Server Controls 都有的属性	174
6.2.3	HtmlAnchor 控件	175
6.2.4	HtmlButton 控件	176
6.2.5	HtmlForm 控件	177
6.2.6	HtmlGeneric 控件	178
6.2.7	HtmlImage 控件	178
6.2.8	HtmlInputButton 控件	179
6.2.9	HtmlInputCheckBox 控件	181
6.2.10	HtmlInputRadioButton 控件	181
6.2.11	HtmlInputText 控件	182
6.2.12	HtmlSelect 控件	182
6.2.13	HtmlTextArea 控件	183
6.2.14	HtmlInputHidden 控件	183
6.2.15	HtmlInputFile 控件	183
6.2.16	HtmlInputImage 控件	185
6.2.17	HtmlTable 控件	186
6.2.18	HtmlTableRow 控件	188
6.2.19	HtmlTableCell 控件	188
6.3	Web Server Controls	189
6.3.1	Web Server Controls 的层次结构	189
6.3.2	Web 服务器端控件都有的属性	190
6.3.3	TextBox 控件	190
6.3.4	Label 控件	192
6.3.5	Image 控件	192
6.3.6	HyperLink 控件	193
6.3.7	Button 控件	193
6.3.8	LinkButton 控件	194
6.3.9	ImageButton 控件	195
6.3.10	DropDownList 控件	196
6.3.11	ListBox 控件	197
6.3.12	CheckBox 控件	200
6.3.13	RadioButton 控件	202
6.3.14	CheckBoxList 控件	203
6.3.15	RadioButtonList 控件	204
6.3.16	Literal 控件	206
6.3.17	Placeholder 控件	207

6.3.18	Panel 控件	207
6.3.19	Table 控件	209
6.3.20	TableRow 控件	211
6.3.21	TableCell 控件	211
6.3.22	AdRotator 控件	211
6.3.23	Calendar 控件	214
6.4	Validation Server Controls	216
6.4.1	简介	216
6.4.2	客户端验证	217
6.4.3	RequiredFieldValidator 控件	217
6.4.4	CompareValidator 控件	218
6.4.5	RangeValidator 控件	220
6.4.6	RegularExpressionValidator 控件	220
6.4.7	ValidationSummary 控件	222
6.4.8	CustomValidator 控件	223
6.5	用户控件	227
6.6	数据绑定	229
6.7	Global.asax 文件	230
6.8	ASP.NET 配置	232
6.8.1	简介	232
6.8.2	Web.config 文件的格式	232
6.9	思考与练习	235
第 7 章	用 ADO.NET 访问数据库	237
7.1	概述	237
7.1.1	.NET Data Provider	237
7.1.2	DataSet	238
7.2	Connection 和 Command 对象	238
7.2.1	Connection 对象	238
7.2.2	Command 对象	240
7.3	DataReader 对象	243
7.3.1	Read() 方法	243
7.3.2	NextResult() 方法	245
7.4	参数查询	245
7.5	DataAdapter 对象	247
7.6	Repeater 控件	249
7.7	DataList 控件	251
7.8	DataGrid 控件	255

7.9 存储过程	274
7.9.1 创建存储过程	274
7.9.2 调用存储过程	275
7.9.3 输出参数	277
7.9.4 使用返回码	279
7.10 思考与练习	281
第8章 综合实例——在线论坛	283
8.1 简介	283
8.2 创建数据库	283
8.3 程序文件	286
8.4 登录及注册	286
8.4.1 登录及登录检查	286
8.4.2 新用户注册	289
8.5 “论坛”主界面	294
8.5.1 index.aspx 程序	294
8.5.2 head.aspx 程序	295
8.5.3 board.aspx 程序	296
8.5.4 display.aspx 程序	299
8.5.5 postthd.aspx 程序	302
8.5.6 getthd.aspx 程序	305
8.5.7 jinhua.aspx 程序	307
8.5.8 banzhu.aspx 程序	311
8.6 注册用户管理	317
附录A 数据库基础	320
A.1 表和数据库	320
A.1.1 表、记录和字段	320
A.1.2 数据库	320
A.1.3 索引	321
A.1.4 关系	321
A.2 创建数据库和表	322
A.2.1 创建数据库	322
A.2.2 创建表	323
A.3 SQL 语言	325
A.3.1 SQL 简介	325
A.3.2 SQL 的构成	326
A.3.3 select 语句	327

A.3.4	delete 语句	328
A.3.5	insert 语句	329
A.3.6	update 语句	329
A.4	ODBC	329
A.5	OLE DB	330
A.6	ADO 对象模型	331

第 1 章 .NET 初步

1.1 什么是.NET

2000 年 6 月,微软公司推出了 .NET 新一代互联网软件和服务战略,引起了信息产业界的广泛关注。.NET 将使微软现有的软件在网络时代不仅适用于传统的 PC 机(个人计算机),而且能够满足呈现强劲增长势头的新设备(如手机、PDA(Personal Digital Assistant)等)的需要。那么,究竟什么是 .NET 呢?请听听微软总裁兼首席执行官 Steve Ballmer 的观点,他说:“.NET 代表了一个集合,一个环境,一个可以作为平台支持下一代 Internet 的可编程结构。”确实,这句话基本上简明扼要地表述了 .NET 的特性。

.NET 首先是一个环境。这是一个理想化的未来 Internet 环境,微软的构想是一个“不再关注单个的网站和与互联网连接的单个设备,而是要让所有的计算机群、相关设备和服务商协同工作的互联网计算环境”。Internet 提供的服务,要能完成更程度的自动化处理。未来的 Internet,应该以一个整体服务的形式展现在最终用户面前,用户只需要知道自己想要什么,而不需要一步步在网上搜索、操作来达到自己的目的。这是一种理想,但确实是 Internet 的发展趋势所在。

.NET 的最终目的就是让用户在任何地方、任何时间,以及利用任何设备都能访问所需要的信息、文件和程序。而用户不需要知道这些东西存放在什么地方,只需发出请求,然后只管接收就是了,而所有后台的复杂性是完全屏蔽起来的。

简而言之,.NET 是一种面向网络、支持各种用户终端的开发平台环境。微软的宏伟目标是让 .NET 彻底改变软件的开发方式、发行方式、使用方式,使软件变成服务。NET 的核心就是要搭建新一代 Internet 平台,用于解决网站之间的协同合作问题,从而最大限度地获取信息。在 .NET 平台上,不同网站之间通过相关的协定联系在一起,网站之间形成自动交流、协同工作,提供最全面的服务。

1.2 Internet 与 Microsoft 平台的演进

Internet 从一开始发展,到目前各类网站如雨后春笋般出现,发展相当迅速,而应用在这方面的技术更是推陈出新,呈现百家争鸣的景象。微软公司为了顺应 Internet 的发展潮流,推出了三代相关的平台。

在第一代 Internet,浏览器只能显示静态的网页。UI(用户接口)逻辑、商业逻辑和数据都存放在后端。应用系统在浏览器与 Web 服务器的帮助下,以类似客户/服务器(Client/

Server)的模式运作。用户从客户端发出 HTTP(超文本传输协议)请求,服务器接收并处理用户的请求后,产生静态的 HTML(超文本标记语言)网页,然后在客户端显示结果。在这一阶段,微软提供了 Internet Explorer 浏览器、IIS(Internet 信息服务)和 COM(组件对象模型)等技术来设计和开发 Web 应用程序。

第二代 Internet 属于 Windows DNA(Distributed Internet Architecture)三层式应用程序结构时代。应用程序分为表示层(Presentation Tier)、商业层(Business Tier)和数据访问层(Data Access Tier)。

- 在表示层,浏览器支持动态网页(DHTML),UI 处理逻辑(如检查身份证号和信用卡号是否有效等)可以直接在浏览器上处理。
- 商业层负责处理商业逻辑,通常会将复杂商业逻辑包装在 COM/COM + 组件中。通过 COM/COM + 处理商业逻辑的组件可集中或分散到几台不同的机器上,再利用群集技术达到扩展性、可靠性及负载均衡(Load Balance)。另外,微软公司还提供了 Web 负载均衡,让几台 Web 服务器可以同时处理用户的请求。
- 数据访问层使用 ADO(ActiveX Data Object)和 OLE DB 等技术提供全局的数据访问模型,以访问各种类型的数据,包括文件、电子邮件、关系数据库和非关系数据库等数据。

第三代 Internet 将是“可编程 Web(Programmable Web)”大放异彩的时代,各个网站与各个系统之间能够更紧密地整合在一起。B2B(Business to Business)、供应链(Supply Chain)、Application Service Provider 等也将真正地落实,而且这些 Web 服务可以在各种不同的便携设备(如手机、PDA 和信息家电等)上执行,真正做到“信息无所不在,随手可得”的境界。为了迎接第三代 Internet 的到来,微软公司推出了 .NET,它建立在 XML(可扩展标记语言)和 Internet 标准协议的基础上,包含了 .NET 的基础结构和基础工具,为开发新型的互动协作软件提供了一个先进的体系结构模型。.NET 有 4 个重要特点:①软件变成服务;②基于 XML 的共同语言;③融合多种设备和平台;④新一代人机界面。这 4 个特点基本上覆盖了 .NET 的技术特征。

1. 软件变成服务

软件从产品形式向服务形式转化,这是整个 IT 行业的大势所趋。在 .NET 中,最终的软件应用是以 Web 服务的形式出现并在 Internet 发布的。Web 服务是一种包装后的可在 Web 上发布的组件。在未来几年内,微软将陆续发布有关 .NET 的平台和工具,用于在 Internet 上开发 Web 服务。到那时,工作在 .NET 上的用户、开发人员和 IT 人员将不再购买、安装和维护软件。取而代之的是,他们将定制服务,软件会自动安装,所有的维护和升级也会通过 Internet 进行。

2. 基于 XML 的共同语言

XML 是源自 SGML(标准通用标记语言)的一种标记语言,它包含一组基本规则,任何人均可用这些规则来创建针对特定应用领域的标记,这些标记不是描述信息的显示方式,而是对信息本身进行描述。良好的数据存储格式、可扩展性、高度结构化及便于网络传输是

XML 的主要特点,也决定了 XML 的卓越性能表现。

在 .NET 体系结构中,XML 作为一种应用间无缝接合的手段,用于多种应用间的数据采集与合并,用于不同应用间的互操作和协同工作。具体而言,.NET 通过 XML 语言定义了简单对象访问协议(SOAP)、Web 服务描述语言(WSDL)、Web 服务发现协议(DISCO)。SOAP 协议是 XML 的实施工具,它提供了一套公共规则集,该规则集说明了如何表示并扩展 XML 数据和命令。WSDL 协议定义了服务描述文档的结构,如类型、消息、端口、端口类型和服务本身等,可以用 WSDL 来表述 Web 服务的具体功能。DISCO 协议定义了如何从资源或资源集中提取服务描述文档、相关服务发现算法等。

3. 融合多种设备和平台

随着 Internet 逐渐成为一个信息和数据的中心,各种设备和服务已经或者正在接入和融入 Internet,成为其中的一部分。.NET 谋求与各种 Internet 接入设备以及相关软件 and 平台的一体化。

4. 新一代人机界面

新一代人机界面主要体现在“智能互动”上。.NET 包括“自然界面”、“通用画布”、“信息代理”和“智能标记”等技术。“自然界面”包括自然语音、视觉、手写等多种模式的输入和表现方法;“通用画布”是一个 XML 复合信息构架,将浏览、通信和资料撰写集成在统一而全面的环境中,使 Internet 从只读环境转换成读写平台,让用户以互动方式撰写、浏览、编辑、注释和分析信息;“信息代理”帮助管理用户在 Internet 上的身份和角色,使用户能更好地控制网络和服务商以何种方式和自己互动;“智能标记”使用户的个人电脑和相关设备能智能地处理来自网上的信息。

1.3 .NET 体系结构

.NET 是从 Windows DNA 平台演进而来的,参见图 1-1。

在开发工具方面,Visual Studio 6.0 演进为 Visual Studio .NET。Visual Studio .NET 是基于 XML 的编程工具和环境,它便于快速开发符合 .NET 体系的软件服务,使其在独立设备、企业数据中心和 Internet 之间的传送更加容易。

在操作系统方面,Windows 2000 演进为 Windows .NET。Windows .NET 紧密地整合了 .NET 的一系列核心构造模块,为数字媒体及应用间协同工作提供支持,是微软的下一代 Windows 平台。

DNA 服务器(DNA Server)演进为 .NET 企业服务器(.NET Enterprise Server)。.NET 企业服务器是企业集成和管理所有基于 Web 的各种应用的基础,提供企业未来开展电子商务的高可靠性、高性能、高可伸缩性及高可管理性。.NET 企业服务器的构成异常庞大而且复杂,共包括表 1-1 所示的各司其职的服务器。

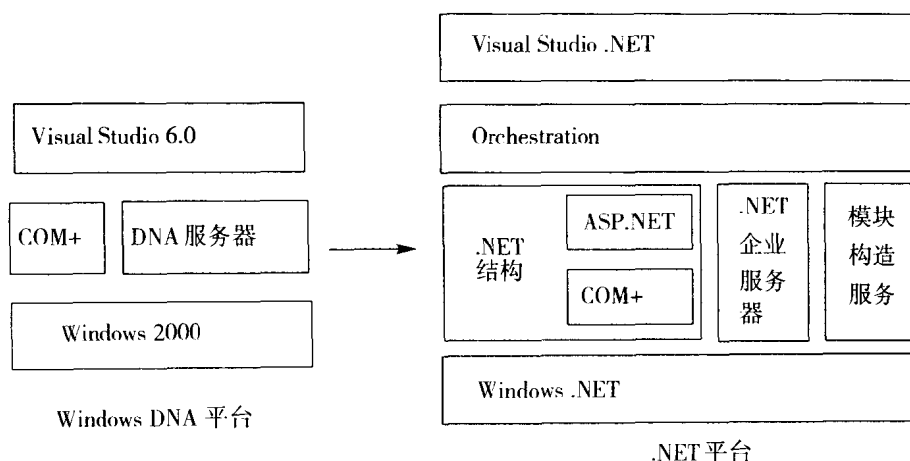


图 1-1 Windows DNA 平台与 .NET 平台

表 1-1 .NET 企业服务器功能简述

.NET 企业服务器	功能描述
Application Center 2000	部署和管理基于 Windows 2000 之上的 Web 应用
BizTalk Server 2000	用于企业间交换商务信息,它利用 XML 作为企业内部及企业间文档传输的数据格式,可以屏蔽平台、操作系统不同的差异,使商业系统的集成成为可能
Commerce Server 2000	用于快速创建在线电子商务
Exchange Server 2000	提供基于 Windows 2000 的通信和协作功能
Host Integration Server 2000	为主机系统的组件集成提供方便
Internet Security & Acceleration Server 2000	主要解决企业应用安全性和可管理性的问题
Mobile Information Server	为移动解决方案提供可靠且具伸缩性的平台
SQL Server 2000	提供完整的数据库和数据分析解决方案

.NET 还增加了许多新的特性,包括 .NET Framework (.NET 框架)、模块构造服务 (Building Block Service) 和 Orchestration 等。其中,.NET 框架的目的是便于开发商更容易地建立 Web 应用程序和 Web 服务,它的关键特色是提供了一个多语言组件开发和执行的环境。

模块构建服务是 .NET 平台中的核心网络服务集合,包括身份认证、目录和搜索、个性化、动态传送、日程安排、XML 数据空间、通知和消息等 Web 服务。Web 开发者可以把这些 Web 服务当成“软件积木”直接组合使用。

Orchestration 主要用在整合跨应用系统、Web 服务及企业的交易流程,它是一种基于 XML 的面向应用的软件集成和自动化处理技术。Orchestration 的目标是尽量不受时间、组织、应用及个人的限制,以便商业事务能交互、动态、可靠地进行下去。Orchestration 有三个基本要求:①处理与执行过程分离,整个处理并不一定非要同执行的细节绑定起来;②动态

处理,随着数据的不同及交换的变化,整个操作过程必须随时动态地更新改变;③整个处理过程不能对参与的平台、应用及协议等做出限制。

1.4 .NET 框架

.NET 框架是一个多语言组件开发和执行环境,它提供了一个跨语言的统一编程环境。.NET 框架的目的是便于开发人员更容易地建立 Web 应用程序和 Web 服务,使得 Internet 上的各应用程序之间,可以使用 Web 服务进行沟通。开发人员可以将远端应用程序提供的服务和单机应用程序的服务结合在一起,组合成一个应用程序。例如,一家购物网站可按这种方式来组成在线商店:使用 Microsoft Passport 服务来验证用户身份,使用合作厂商的信用卡处理服务及自行开发的产品服务等。从层次结构来看,.NET 框架又包括三个主要组成部分:公共语言运行时 (CLR: Common Language Runtime)、服务框架 (Services Framework) 和上层的两类应用模板——传统的 Windows 应用程序模板 (Win Forms) 和基于 ASP.NET 的面向 Web 的网络应用程序模板 (Web Forms 和 Web Services),参见图 1-2。

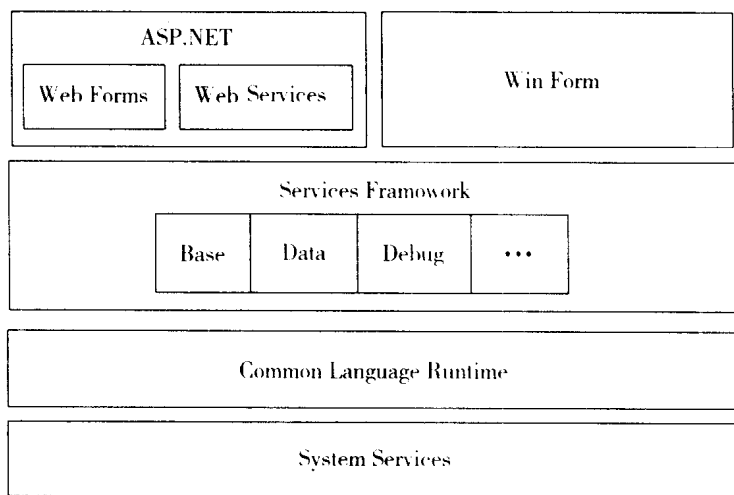


图 1-2 .NET 框架

1.4.1 公共语言运行时

.NET 提供了一个运行时环境,叫做公共语言运行时 (CLR)。CLR 管理代码的执行并使开发过程变得更加简单。CLR 是一种受控的执行环境,其功能通过编译器与其他工具共同展现。以“运行时”为目标的代码称为受控代码 (managed code)。受控代码指向的对象在执行过程中完全被 CLR 所控制。在执行过程中,CLR 提供自动内存管理、调试支持、增强的安全性及与非受控代码 (如 COM 组件) 的互操作性。凡是符合 CLS (公共语言规范) 的程序语