

信息科学与技术丛书

杨云 刘君 编著

精通 ASP.NET 4.0

- 全面讲述ASP.NET 4.0
- 深入分析开发环境和体系结构
- 介绍各种控件、页面主题/皮肤及配置
- 开发电子商务交易系统
- 开发博客系统



 机械工业出版社
CHINA MACHINE PRESS

信息科学与技术丛书

精通 ASP.NET 4.0

杨 云 刘 君 编著



机械工业出版社

本书采用知识讲述+代码示例的方式，全面讲述了 ASP.NET 4.0 的各个方面。主要内容包括：ASP.NET 4.0 的开发环境、体系结构、各种控件、页面主题/皮肤、配置以及站点国际化。还详细介绍了如何开发电子商务交易系统 and 博客系统。全书力求帮助读者迅速掌握 ASP.NET 4.0 程序的设计方法，应用 ASP.NET 完成编程任务。

本书适合网站开发人员及相关专业师生阅读。

本书代码可在 <http://www.cmpbook.com> 免费下载。

图书在版编目（CIP）数据

精通 ASP.NET 4.0 / 杨云，刘君编著. —北京：机械工业出版社，2013.1

（信息科学与技术丛书）

ISBN 978-7-111-41437-7

I. ①精… II. ①杨… ②刘… III. ①网页制作工具—程序设计

IV. ①TP393.092

中国版本图书馆 CIP 数据核字（2013）第 026060 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）

策划编辑：车 忱

责任编辑：车 忱

责任印制：乔 宇

三河市国英印务有限公司印刷

2013 年 3 月第 1 版·第 1 次印刷

184mm×260mm·27.5 印张·682 千字

0001—3500 册

标准书号：ISBN 978-7-111-41437-7

定价：69.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务

网络服务

社服务中心：（010）88361066

教材网：<http://www.cmpedu.com>

销售一部：（010）68326294

机工官网：<http://www.cmpbook.com>

销售二部：（010）88379649

机工官博：<http://weibo.com/cmp1952>

读者购书热线：（010）88379203

封面无防伪标均为盗版

出版说明

随着信息科学与技术的迅速发展，人类每时每刻都会面对层出不穷的新技术和新概念。毫无疑问，在节奏越来越快的工作和生活中，人们需要通过阅读和学习大量信息丰富、具备实践指导意义的图书来获取新知识和新技能，从而不断提高自身素质，紧跟信息化时代发展的步伐。

众所周知，在计算机硬件方面，高性价比的解决方案和新型技术的应用一直备受青睐；在软件技术方面，随着计算机软件的规模和复杂性与日俱增，软件技术不断地受到挑战，人们一直在为寻求更先进的软件技术而奋斗不止。目前，计算机和互联网在社会生活中日益普及，掌握计算机网络技术和理论已成为大众的文化需求。由于信息科学与技术 in 电工、电子、通信、工业控制、智能建筑、工业产品设计与制造等专业领域中已经得到充分、广泛的应用，所以这些专业领域中的研究人员和工程技术人员越来越迫切需要汲取自身领域信息化所带来的新理念和新方法。

针对人们了解和掌握新知识、新技能的热切期待，以及由此促成的人们对语言简洁、内容充实、融合实践经验的图书迫切需要的现状，机械工业出版社适时推出了“信息科学与技术丛书”。这套丛书涉及计算机软件、硬件、网络 and 工程应用等内容，注重理论与实践的结合，内容实用、层次分明、语言流畅，是信息科学与技术领域专业人员不可或缺的参考书。

目前，信息科学与技术的发展可谓一日千里，机械工业出版社欢迎从事信息技术方面工作的科研人员、工程技术人员积极参与我们的工作，为推进我国的信息化建设作出贡献。

机械工业出版社

前 言

读者在学习本书前应该明确，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的精力投入到改善设计和完善用户体验上。ASP.NET 作为微软主力开发技术经历了好几个版本，微软在不断总结和听取反馈后发布了 ASP.NET 4.0。

ASP.NET 4.0 继承了 ASP.NET 一贯的编程模式、代码设计、实现方法和语法模型。变化最大的就是 ASP.NET 4.0 简化了开发过程，给予设计人员更多思考软件设计的时间。

请读者注意的是，ASP.NET 4.0 比 ASP.NET 2.0/3.5 开发的项目减少了 20%~30% 的人工编码量，更多的代码由 IDE 自动生成。ASP.NET 4.0 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，如图形控件、用户登录、用户创建、用户管理、WebPart 和数据源控件等。

ASP.NET 4.0 和之前的版本相比更加兼容 XHTML 标准，控件的呈现结果将根据客户端的设备不同而呈现不同的内容并支持 W3C 标准。

ASP.NET 4.0 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 4.0 还增强了缓存管理等功能。

本书共 15 章，主要面向使用 C# 4.0 开发 ASP.NET 4.0 应用的开发人员，深入讲解了 ASP.NET 4.0 的运行原理、控件使用、C# 4.0 基础知识和时尚的图形控件技术，力争使读者通过学习能掌握如何使用 VS2010 开发基于 ASP.NET 4.0 的应用。

本书对于比较重要的理论知识都安排有相应的短小实例代码进行讲解。读者可以按照书中的示范编写代码来巩固知识点。

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。在本书第 14 和 15 章安排了两个流行的实例讲解，在实例的讲解中都采用了精讲的方法，力求用足够的篇幅将本书的知识点进行串接，更好地帮助读者梳理所学知识。

本书的读者不要求必须有 ASP.NET 2.0/3.5 的知识，本书既适合 ASP.NET 4.0 的初学者和 ASP.NET 2.0/3.5 开发人员进行技术升级，也可作为大中专院校相关专业教材使用。

致谢——

感谢母亲对我无微不至的照顾和支持。

杨 云
2012.05

目 录

出版说明

前言

第 1 章 .NET 简介	1	2.9.3 快速信息	26
1.1 .NET Framework 4.0 在.NET		2.9.4 自动完成	26
技术体系中的位置	1	2.9.5 C#相关的智能感知	27
1.2 .NET 4.0 各部分的功能	2	2.10 对重构的支持	29
1.3 .NET 4.0 的组件	3	2.11 调试和测试	31
1.3.1 Windows Presentaion		2.12 页面与代码的组织	32
Foundation	3	2.13 ASP.NET 4.0 应用程序	
1.3.2 Windows Communication		文件夹	35
Foundation	5	2.14 ASP.NET 4.0 的预编译	39
1.3.3 Workflow Foundation	5	第 3 章 ASP.NET 4.0 体系结构	43
1.4 搭建.NET 4.0 的开发环境	6	3.1 代码模型	43
1.4.1 在 Windows XP/2008/Win7 上		3.2 代码的结构	44
搭建开发环境	6	3.3 编译模型	45
1.4.2 相关工具	9	3.4 扩展性与管道技术	46
第 2 章 Visual Studio.NET 2010		3.5 缓存技术	47
开发环境	12	第 4 章 ASP.NET 4.0 网络服务	50
2.1 安装 VS2010	12	4.1 网络服务 (Web Service)	
2.2 创建和打开 Web 站点	12	基础	50
2.3 使用内置的 ASP.NET		4.1.1 Web Service 的概念	50
Deployment Server	14	4.1.2 Web Service 的基础技术	50
2.4 迁移现有的 VS2005/VS2008		4.1.3 Web Service 的软件支持	51
Web 站点	15	4.1.4 Web Service 的编码模型	51
2.5 编辑 Web 站点	18	4.1.5 使用 Visual Studio 2010	
2.6 使用服务器控件	21	开发 Web Service	52
2.7 创建事件处理程序	22	4.2 Web Service 的演进方向	54
2.8 验证 HTML 源码的可用性	23	4.3 基于接口的服务约定	55
2.9 使用 Visual Studio 的		4.4 更多的 XSD/WSDL 改进	57
Intellisense	25	4.5 更好的互操作性	57
2.9.1 列出对象成员	25	4.6 为 Windows Communication	
2.9.2 显示方法参数信息	26	Foundation 做好准备	59

第 5 章 ASP.NET 4.0 功能

增强控件	61
5.1 图表控件	61
5.2 数据源控件	64
5.2.1 SqlDataSource 数据源控件	65
5.2.2 XmlDataSource 数据源控件	68
5.2.3 ObjectDataSource 数据源控件	71
5.2.4 AccessDataSource 数据源控件	71
5.2.5 SiteMapDataSource 数据源控件	72
5.3 GridView 控件	72
5.3.1 使用 GridView 显示数据	72
5.3.2 使用自定义数据列	77
5.3.3 使用模板列	80
5.3.4 删除数据	82
5.3.5 控件参数	86
5.3.6 利用数据源控件缓存数据	88
5.4 DetailsView 控件	88
5.4.1 使用 DetailsView 显示、编辑 和删除数据	89
5.4.2 插入新记录	92
5.4.3 使用模板	92
5.4.4 同时使用 GridView 和 DetailsView	96
5.5 TreeView 控件	97
5.5.1 使用静态数据	97
5.5.2 使用动态数据	98
5.5.3 通过数据库填充控件	99
5.6 Login 控件	101
5.7 PasswordRecovery 控件	102
5.8 LoginStatus 和 LoginName 控件	103
5.8.1 LoginStatus 控件	103
5.8.2 LoginName 控件	104
5.9 LoginView 控件	104
5.10 CreateUserWizard 控件	105
5.11 BulletedList 控件	107
5.12 ImageMap 控件	109

5.13 MultiView 和 View 控件	111
5.14 Wizard 控件	113
5.15 Panel 控件	115
5.16 FileUpload 控件	117
5.17 HiddenField 控件	118
5.18 Substitution 控件	119

第 6 章 ASP.NET 4.0 中的

MasterPager

6.1 新建 MasterPager	121
6.2 在内容页嵌入 MasterPager	122
6.3 使用多个内容区域和默认 内容	123
6.4 动态使用 MasterPager	127
6.5 在运行时访问 MasterPager	129
6.6 嵌套的 MasterPager	131

第 7 章 ASP.NET 4.0 成员和

角色管理

7.1 认证和授权	134
7.1.1 IIS 和 ASP.NET 用户认证流程	134
7.1.2 认证	135
7.1.3 授权	135
7.2 ASP.NET 4.0 用户认证	135
7.2.1 使用 ASP.NET 管理工具 添加用户	138
7.2.2 使用 CreateUserWizard 创建 用户	140
7.2.3 改变默认的 Provider 设置	141
7.2.4 个性化 CreateUserWizard 控件	141
7.2.5 使用 Login 相关的控件	143
7.3 ASP.NET 角色管理系统	147
7.3.1 角色管理	147
7.3.2 角色管理和成员管理的关系	148
7.3.3 应用角色管理	148
7.3.4 修改<RoleManager>节点	150
7.3.5 使用用户角色控件	151
7.4 使用 Membership/Role API	153
7.4.1 使用 Membership API 管理用户	153

7.4.2 使用 Role API 进行用户 角色管理	155	信息通信	196
7.5 ASP.NET 的 Membership Provider	158	9.2 个性化 WebPart 的数据存储 和转移	198
7.5.1 SqlMembershipProvider	159	第 10 章 创建 ASP.NET 服务器控件	201
7.5.2 ActiveDirectoryMembership Provider	160	10.1 ASP.NET 服务器控件概述	201
7.6 实现自定义的 Membership Provider	162	10.2 服务器控件项目的设置	205
7.7 基于角色的站点导航	166	10.3 服务器控件的呈现	206
第 8 章 窗体页设计技巧	170	10.3.1 输出控件的内容	207
8.1 Page 类的新事件	170	10.3.2 为 HTML 元素添加属性	207
8.2 添加标题	171	10.3.3 控件的适应性	208
8.3 设置焦点	172	10.4 开始创建服务器控件	210
8.4 为 Form 设定默认按钮	173	10.5 创建复合控件	217
8.5 更好的输入验证控件	173	10.6 为控件添加更多功能	227
8.6 使用 Page.Items 字典	176	10.6.1 为控件添加输入验证	227
8.7 使用跨页面传送功能	176	10.6.2 控件的子属性	229
8.8 高速缓存和 SQL Server Invalidation 功能	179	10.6.3 为 Register 控件增加嵌 套子属性	231
8.9 配置 SQL Server Invalidation	179	10.7 控件的回调示例——异步 请求	233
8.10 使用 SQL Server Invalidation 和数据源控件	180	第 11 章 ASP.NET 4.0 中的页面 主题/皮肤	237
8.11 通过编程方式使用 SQL Server Invalidation	181	11.1 页面主题概述	237
8.12 高速缓存的其他改进	182	11.2 页面主题的运用	238
8.13 使用页面高速缓存	182	11.2.1 App_Themes 目录	238
第 9 章 使用 ASP.NET 4.0 Web Part 框架	183	11.2.2 全局页面主题和局部 页面主题	239
9.1 常用 WebPart 控件	183	11.3 皮肤文件和主题的使用	240
9.1.1 WebPartManager 控件	183	11.4 使用样式表主题	250
9.1.2 WebPartZone 控件	185	11.5 资源与主题	252
9.1.3 CatalogZone 控件和所属 CatalogPart 控件	188	11.6 动态加载页面主题	254
9.1.4 EditorZone 和 所属 EditorPart 控件	193	第 12 章 ASP.NET 4.0 配置详解	258
9.1.5 ConnectionsZone 控件和		12.1 ASP.NET 配置的基本结构	258
		12.1.1 .NET 应用程序的配置体系	258
		12.1.2 ASP.NET 配置结构	258
		12.1.3 .NET 配置文件基本结构	259
		12.1.4 配置区域和配置组	259

12.1.5 添加自定义的配置节	261	14.1.2 系统业务流程设计	294
12.1.6 使用 location 节点和 path 属性	262	14.2 系统架构与功能模块	297
12.1.7 ASP.NET 常用配置节点	263	14.3 数据库设计与实现	313
12.2 获取配置信息	266	14.3.1 数据库需求分析	313
12.3 使用 ASP.NET 配置 管理接口	269	14.3.2 数据表设计	315
12.3.1 使用配置管理接口访问 程序配置	269	14.3.3 存储过程设计	333
12.3.2 对配置内容加密	270	14.4 用户交互处理层设计与实现	337
12.4 使用 ASP.NET 配置工具	272	14.4.1 用户交互处理层结构	337
12.4.1 使用 ASP.NET 管理控制台	272	14.4.2 系统的主题	339
12.4.2 使用 ASP.NET 管理站点	274	14.4.3 ASP.NET AJAX 技术的运用	340
12.4.3 使用 ASPNET_REGSQL 工具	275	14.4.4 电子结算模块	342
12.4.4 使用 ASPNET_REGIIS 工具	276	14.4.5 用户自定义控件	348
12.5 ASP.NET 页面配置	276	14.4.6 母版页	357
12.6 配置 ASP.NET 进程模型	278	14.4.7 普通功能页	361
第 13 章 站点的国际化和本地化	280	第 15 章 开发博客系统	378
13.1 国际化和本地化	280	15.1 系统概述	378
13.1.1 什么是国际化和本地化	280	15.1.1 系统需求分析	378
13.1.2 ASP.NET 4.0 对国际化的支持	281	15.1.2 系统业务流程设计	380
13.2 自动检测浏览器语言	281	15.2 系统架构与功能模块	382
13.2.1 在浏览器中设置语言偏好	281	15.3 数据库设计与实现	386
13.2.2 使 ASP.NET 页面能够自动检测 浏览器语言文化设定	281	15.3.1 数据库需求分析	386
13.3 ASP.NET 程序中的本地化	284	15.3.2 数据表设计	387
13.3.1 无代码本地化	284	15.3.3 存储过程设计	389
13.3.2 从代码中访问资源文件	290	15.4 用户交互处理层设计与实现	393
第 14 章 开发电子商务交易系统	291	15.4.1 用户交互处理层结构	393
14.1 系统概述	291	15.4.2 多语言本地化	394
14.1.1 系统需求分析	291	15.4.3 用户自定义控件	395
		15.4.4 系统母版页	404
		15.4.5 普通功能页	406
		参考文献	431

第 1 章 .NET 简介

随着 Windows 7 的发布，微软 .NET Framework 也发展到了一个新的阶段。Microsoft .NET Framework 4.0 是 .NET Framework 平台的一个新的里程碑。它可以在 Vista、Windows 7 和 XP 上运行。

Microsoft .NET Framework 4.0 是微软 .NET 软件平台的最新发展成果。它基于 .NET 4.0 运行时环境，并整合了 .NET 4.0 新增的多种功能，如 Windows Presentation Foundation（Windows 界面基础框架），Windows Communication Foundation（Windows 通信基础构架），Workflow Foundation（工作流基础构架）和 Windows CardSpace，并对 C# 和 VB 等语言进行了增强。

本章对 .NET Framework 4.0 及其组件进行了整体描述，目的是使读者对这一新版本有一个清晰的了解，同时分析了采用的技术，并给出了较为详细的说明。

1.1 .NET Framework 4.0 在 .NET 技术体系中的位置

与 .NET 4.0 给开发者带来大量新鲜而强大的类库（WCF/WPF/WF）不同的是，.NET Framework 4.0 带来了语法上的大幅增强，提供了新的编译器，优化了 .NET Framework 底层运行时，并对 WPF/WCF/WF 这几个重要的组成部分进行了升级和优化。虽然如此，.NET Framework 4.0 仍然是基于 .NET 的公共语言运行时框架（CLR）。本书重点介绍的内容如图 1-1 所示。

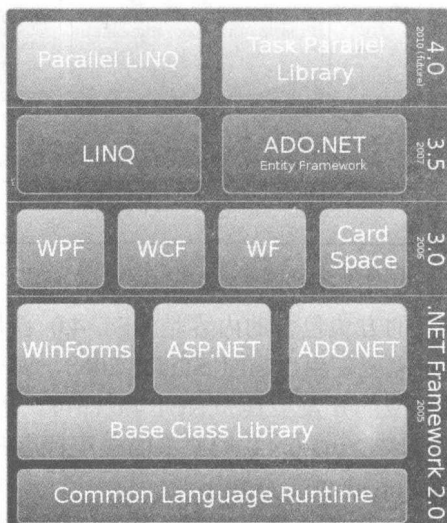


图 1-1 .NET Framework 4.0 的结构

由图 1-1 可见，.NET 4.0 并未对 .NET 现存的技术进行任何改动，包括 ASP.NET，

ADO.NET 和 WinForms 在内的主要技术都保持原样，这对熟练掌握.NET 的技术人员是个好消息，因为他们所掌握的技术仍然很有价值。从微软.NET 平台的发展趋势来看，.NET 平台将为.NET 4.0 提供基础类库，4.0 将着重引入语法、数据访问和 Web 客户端上的一些创新。因此可以把.NET 4.0 看做.NET 的超集，是对.NET 的一次补充。

如果开发者从.NET 2.0 或者 3.5 迁移到 4.0，那么需要考虑代码的兼容性问题。尽管.NET 框架已经尽力保证向后兼容，但是由于一些安全方面的改进，仍然有少数重要操作存在不兼容的现象。不过从.NET 3.5 到 4.0 则完全不存在这个问题，因为.NET Framework 3.5 的所有组件都可在支持.NET Framework 4.0 的平台上运行。

从上面的描述中，可以得出一个推论，即.NET 4.0 是承上启下的一代，它继承了.NET 3.5 强大的基础平台，使.NET 平台在用户界面、网络通信、工作流和身份管理等方面彻底进入了一个新的时代。.NET 4.0 在 3.5 的基础上增加了 LINQ（语言整合查询）和 ASP.NET 的 OR/M 框架（ASP.NET Entity Framework）等新功能，这些新功能极大地增强了 VB 和 C# 等语言，简化了开发工作，它们可以和 WPF/WCF 和 WF 共同工作，来创建强大的应用程序。

1.2 .NET 4.0 各部分的功能

.NET 4.0 的基本结构在图 1-1 中已经比较清晰地展现出来了。图 1-2 更清晰地显示了.NET 4.0 相关的开发组件之间的层次结构。

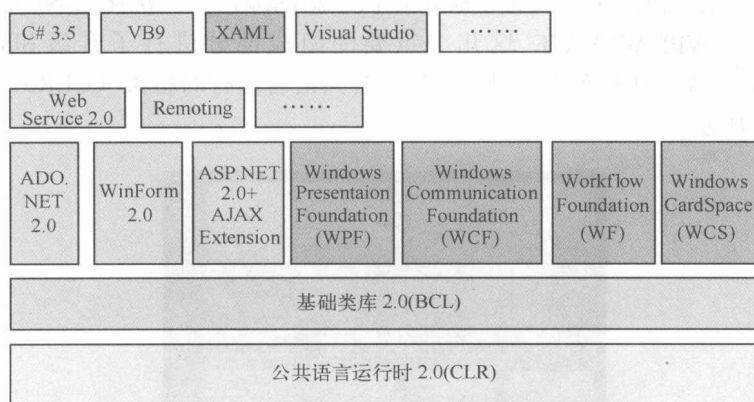


图 1-2 .NET 4.0 组件关系图

如图 1-2 所示，较深颜色的方块包含的内容是.NET 4.0 新增的，其他的都继承自.NET 2.0。开发.NET 3.5 的应用程序需要使用 Visual Studio 2008，而开发.NET 4.0 仍然可以使用 Visual Studio 2008，只要安装相应的开发工具和.NET 4.0 即可。也可以直接安装 Visual Studio 2010，Visual Studio 2010 直接支持 2.0/3.0/4.0 框架下的应用程序开发。

(1) Windows Presentation Foundation (WPF)

WPF 是微软潜心开发的新一代用户界面框架。Windows 系统的用户界面的历史可以从 Windows 1.0 开始计算，期间经历了 GDI、GDI+，直到 WinForms，用户界面技术一直都没有得到根本性的发展，只不过可以看做一次次的重构和封装，其本质依然是基于软件模拟渲染，不支持硬件图形加速。与此同时，Windows 游戏的图形界面 DirectX 则是飞速地发展，体积阴影、高精

度纹理、硬件光源等技术的大规模应用，让 Windows 游戏世界一步一步向乱真的方向发展。桌面应用程序和游戏的图形效果差距过大，显卡的图形加速功能被普通应用程序所浪费，以及用户对一成不变的界面渐渐厌倦等原因，让微软决心推出新一代的用户界面框架，让普通应用程序的用户界面也能利用 DirectX 中的图形特效，并充分利用图形卡的加速功能，带给用户更好的体验。

(2) XAML (eXtensible Application Markup Language: 可扩展应用程序标记语言)

XAML 是为 WPF 量身定做用于用户界面描述的标记语言。XAML 采用了 XML 的格式，易于阅读，结构规范，也可充分利用 XML 的强大扩展性与其他图形绘制工作进行交互。

(3) Windows Communication Foundation (WCF)

在 .NET 4.0 之前，Windows 操作系统下存在数种分布式消息交换的技术，如 MSMQ、Web Service、Remoting、企业服务 (Enterprise Services)、WSE 等，这些技术涉及的编程模型也千差万别。微软创建 WCF 的一个目的就是统一 Windows 平台的分布式通信技术，让这些技术以统一的模型对外提供服务。这种统一的模型让 .NET 框架成为了更完善的面向服务 (SOA) 的平台。

(4) Workflow Foundation (WF)

工作流基础类库是新增加的大型基础类库。微软在此之前从未向外界提供过类似的开发包，因此这也是一次尝试。根据笔者的观察和试用，该开发平台相当强大，足够作为工作流引擎的基础平台。

1.3 .NET 4.0 的组件

1.3.1 Windows Presentaion Foundation

Windows Presentaion Foundation (WPF) 是一个全新的 UI 体系结构，它不仅能比以往的 UI 构架做得更多，还能做得更好、更容易。在 WPF 中可以发现多种用户界面技术的痕迹，如 GDI 和 GDI+，这一点其实毫无疑问，毕竟 WPF 是 GDI、GDI+ 的接班人。又如 HTML，WPF 引入了 XAML 语言作为界面描述语言，显然受到了 HTML 广泛应用的影响。在引入 XAML 进行 UI 描述以后，Windows 程序就可以采用 ASP.NET 那样的代码后置，从而将界面和程序逻辑分离。WPF 处理动画的方式显然吸取了 Flash 时间线 (Timeline) 的优点并且发展出独特的 StoryBoard 系统。最后，必须提到的是，WPF 为用户界面的 3D 化提供了强大的支持。在 WPF 之前，如果要在用户界面中提供 3D 元素，必须采用 2D 模拟 3D 的办法，或者采用 DirectX/OpenGL 渲染的方式。这两种方式存在的问题是：2D 模拟 3D 的性能非常低，不可能为 3D 元素提供多少特效；采用 DirectX 或者 OpenGL 模式实现的用户界面不容易与其他 Windows 界面元素进行交互。但在 WPF 中，这些都不再是问题，因为 WPF 采用了 DirectX 9.0C 来渲染 3D 元素，不仅原生地支持了 3D 元素，而且由于采用硬件加速，因此大大提升了用户界面的显示效率，使得开发人员有机会为用户界面提供更多、更酷的效果。

WPF 吸取了多种技术的优点创造出一个新型的平台，在应用程序类型的支持种类上也比它的前辈们有不少提高。除了传统的 Windows 窗口程序以外，WPF 还支持浏览器应用程序 XBAP，XBAP 可以在客户端的浏览器中运行，它受到浏览器的安全性限制；WPF 还吸收了 Web 程序的特点——功能按页面进行分离，新增了 Navigation Application (浏览型应用程序)，可以让用户在同一个窗体中，如同访问 Web 页面那样执行程序，这个特性将会在后面

的章节中进行较为详细的讲解。

注意：WPF 还有一个子集，叫做 Silverlight（开发代号为 WPF/Everywhere），它将作为一个浏览器的客户端技术运行在多个平台的浏览器中。在本书开始编写时，Silverlight 的 1.0 版本的 beta1 才公布不久，因此本书就不对它进行更多的说明。感兴趣的读者可以自行查阅微软站点的资料。

WPF 虽然是一个全新的 UI 框架，不过为了在最大程度上使 WinForms 的知识得以保留，开发人员仍然可以用相似的代码结构来创建窗体和控件。下面的两段代码分别使用了 WinForms 和 WPF 技术来创建一个窗体和一个按钮。WinForms 代码如下所示：

```
using System.Windows.Forms;
using System;

class program
{
    [STAThreadAttribute()]
    public static void Main()
    {
        Form form = new Form();
        form.Text = "Winform";
        Button btn = new Button();
        btn.Text = "click me";
        btn.Location = new System.Drawing.Point(100, 70);
        form.Controls.Add(btn);
        Application.Run(form);
    }
}
```

WPF 代码如下所示：

```
class program : Application
{
    [STAThreadAttribute()]
    public static void Main()
    {
        program app = new program();
        Window window = new Window();
        window.Title = "WPF Window";
        Button btn = new Button();
        btn.Width = 100; btn.Height = 40;
        btn.Content = "Click me";
        window.Content = btn;
        app.Run(window);
    }
}
```

两段代码的结构大同小异，分别定义了一个 Form 和 Window 对象作为主窗口，然后在主窗口中添加了一个按钮对象。稍微有所不同的地方是，在 WPF 中，如果不定义按钮的长和宽，那么在默认情况下，按钮会充满整个包含它的容器。读者可以尝试执行去掉 btn.Width = 100; 和 btn.Height = 40; 这两个语句，看看窗口的效果。WinForms 和 WPF 在表面上看起来非常相似，但是它们却有着本质上的不同。WinForms 技术实际上只是 GDI 的托管封装，内部仍然使用相同的控件和消息模型。

除了使用 C# 和 VB.NET 等语言进行编程以外，还可使用上文已经提到的 XAML 对窗口进行描述，下面的代码是采用 XAML 标记语言编写的，它的功能与上面的代码相同，用来描述一个窗口：

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="WPFStart" Height="300" Width="300">
    <Button Content="Click Me" Height="40" Width="100"/>
</Window>
```

代码总体感觉和 HTML 比较类似。Window 标签用于声明一个 Window 对象，xmlns 的作用是声明一个 XML 的命名空间，xmlns:x 也是相同的作用，关于它们的详细解释可以查看第 4 章。<Button/> 标签用于声明一个按钮对象，并且通过按钮对象的 Content 属性指定按钮上显示的内容，可以指定除文字以外的内容，如图片、视频、其他控件容器等，甚至可以指定一个按钮，基本上能用在 UI 上的元素都可以指定。

关于 WPF 更多的内容，将从第 3 章开始系统介绍。

1.3.2 Windows Communication Foundation

Windows Communication Foundation (WCF) 除了建立一个统一的分布式通信编程模型以外，还有三个重要的目标：

- 基于面向服务的构架，并成为面向服务的基础平台。
- 以更健壮的方式实现 Web Service 和 .NET Remoting。
- 全面实现 Web Service 的标准协议簇 WS-*。

微软创建 WCF 来为 .NET 平台提供一种统一的分布式计算平台，它应当具有广泛的互操作性以及对面向服务构架的直接支持。

微软以及许多领先 IT 企业已经开始为面向服务(SOA)的平台作好了准备，WCF 就是微软公司的 SOA 战略中重要的一环。

面向服务的应用程序和先前的分布式应用程序有显著的不同。面向服务的应用程序的服务端和客户端是松散耦合的，客户端和服务端不必事先知道对方的存在，只要通过一个 URL，按照约定的通信方式就可以进行互操作。

读者或许会有一些疑问：在 .NET 环境中，使用 Visual Studio、.NET Framework 以及 WSE 可以非常容易地编写 Web Service 来创建服务端或者客户端，并且可以和运行在其他平台上的客户端或服务端进行交互。那么为什么还需要 WCF？

如开发者所知，Web Service 只是 .NET 平台上其中一种可以用来构建分布式应用程序的技术，其他提到过的技术还包括 Remoting、Enterprise 和 MSMQ 等。运用这些技术来创建分布式服务的方式和互操作性有很大的区别，而且当试图转换分布式计算的基础到另一种技术时，代价将相当大。WCF 就是要解决这个问题，它统一了 .NET 环境下分布式计算的编程模式，让开发人员可以不用太关心基础的分布式计算的机制。WCF 不仅可以编写 Web 环境可以访问的服务，而且可以在其他协议的网络中部署服务。

1.3.3 Workflow Foundation

Workflow 即工作流，对大多数人来说都是比较新鲜的概念。但实际上，许多软件都涉

及工作流的处理。在 Workflow Foundation (WF) 中, 工作流就是用一系列抽象出来的活动 (Activities) 来描述现实世界中的一个流程。在现实中, 工作流往往代表着一件工作从一个人传递到另一个人, 并伴随着状态的改变。一个典型的工作流往往定义了几个必要的步骤, 步骤可以是有顺序的, 也可以是无顺序的。举例来说, 一个员工报销医疗保险费用, 他的流程就必须是: 填写理赔单→提交表单→公司审核→医保机构审核→发放理赔金额, 这个流程的每个步骤都是必须而且有序的, 到最终发放理赔金额之前, 任何一个步骤的失败都将导致流程中断。这个工作流比较简单, 路径非常单一, 而复杂的工作流则涉及更多的分支。在大多数程序中, 涉及的过程处理都可以看作广义上的工作流。

工作流是由活动元素组成的, 可以将工作流拆解成为一个一个的活动 (Activity) 对象, 然后按照逻辑组织到一起描述工作流。

目前, 大多数软件对工作流的处理方式都是私有和不开放的, 缺乏共通性。不同的软件定义的工作流无法重用和扩展, 这对软件的快速开发显然是极为不利的。WF 就是将创建一个工作流所需要的各个环节抽象出来, 供开发者使用。WF 并不关心开发者使用 WF 搭建的工作流是用来处理什么具体事务的, 可以说只要处理的事务带有流程的性质, 就可以应用 WF。

在 WF 中, 工作流可以分为两种, 一种是顺序流, 另一种是基于状态机的工作流。那么, 在什么样的情况下使用哪一种工作流呢? 通常情况下, 如果一个流程重点关注的是顺序, 较少涉及对象状态的变化, 则多用顺序流; 如果流程非常依赖于对象的状态, 而对象的状态在整个流程中会发生多次变化, 则基于状态机的工作流将是更好的选择。

1.4 搭建.NET 4.0 的开发环境

.NET 4.0 是随着 Visual Studio 2010 一起发布的。使用 Visual Studio 2010 可以很自然地开发 .NET 4.0 应用程序。下面分别讲解如何在 Windows XP、Windows 2008 以及 Win7 系统上搭建开发环境。

1.4.1 在 Windows XP/2008/Win7 上搭建开发环境

由于在 Vista 和 Windows 2008 上搭建开发环境的步骤基本相同, 因此在此一道进行讲述。搭建开发环境的前提条件是:

(1) 操作系统必须是 Windows Vista、Win7 家庭版或者专业版, 并且是 SP2 之后的版本 (包括 SP2), 或者是 Windows 2008 任一版本的 SP1 之后的版本 (包括 SP1)。

(2) 用户的计算机必须具有下面的硬件配置才能保证 .NET 4.0 应用程序的运行:

最低: 奔腾 400MHz CPU 96MB 内存

推荐: 奔腾 1GHz 以上 CPU 512MB 及以上内存

(3) 用户已经安装了 Express 版以外的任一版本 Visual Studio 2010。如果没有安装, 则必须首先安装 Visual Studio 2010 才可以继续后面的步骤。

在具备上面的这些条件以后, 就可以安装 .NET 4.0 开发环境了。可以参考如下步骤进行安装:

1) 从微软网站上下载 Microsoft .NET Framework 4.0 Redistributable Package。

2) 双击下载的安装包进行 .NET Framework 4.0 的安装。根据安装提示进行安装即可。

3) 双击获取的 .NET Framework 4.0 SDK 进行安装, 安装过程中可以选择安装的路径。

妥善选择 SDK 将要安装的驱动器，确保驱动器有足够的空间。

4) 安装 Visual Studio 2010。

在 Vista 上搭建开发环境，与在 XP/2008 系统上大同小异。安装完成以后，打开 Visual Studio 2010，在启动过程中出现如图 1-3 所示的安装组件。

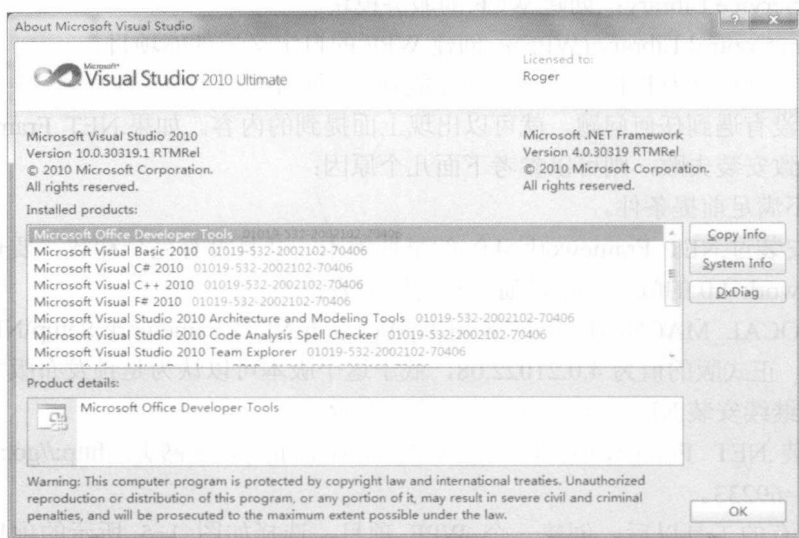


图 1-3 安装组件

与此同时，打开 Visual Studio 2010 的文件菜单，单击创建新项目，可以发现 Visual C# 和 Visual Basic 应用程序中多了一个下拉菜单，菜单选项包括 .NET Framework 2.0、.NET Framework 3.5 和 .NET Framework 4.0，如图 1-4 所示。



图 1-4 新的应用程序类型

可以看到，.NET Framework 4.0 项目类型中增加了几个新的应用程序类型：

1) Windows Application (WPF)：使用 WPF 类库来创建窗体应用程序的项目类型。

2) XAML Browser Application(WPF)：使用 WPF 类库来创建运行在浏览器中的应用程序的项目类型。

3) WCF Service Library：创建 WCF 的服务模块。

4) Custom Control Library (WPF)：创建 WPF 的自定义控件库项目。

如果安装有其他开发模板，程序类型可能有更多的不同。

如果安装没有遇到任何问题，就可以出现上面提到的内容。如果.NET Framework 4.0 安装出现错误导致安装失败，则可以参考下面几个原因：

1) 系统不满足前提条件。

2) 之前安装过.NET Framework 4.0 的早期版本，并且没有完全卸载。要查看之前安装的.NET Framework 4.0 的版本，可以通过查看注册表：

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\NET Framework Setup\NDP\v4.0\Setup 的 Version 值，正式版的值为 4.0.21022.08，低于这个版本可以认为是预发布版本，必须完全卸载以后才能继续安装.NET Framework 4.0 的正式版。

有关安装.NET Framework 4.0 更详细的信息请参阅网址 <http://go.microsoft.com/fwlink/?LinkId=69233>。

安装好所有的工具以后，创建一个 WPF 项目，选择如图 1-5 所示的项目类型，单击“确定”按钮完成创建，此时，Visual Studio 2010 会呈现出如图 1-6 所示的界面。



图 1-5 选择创建新项目类型