

<<C#高级编程>>

图书基本信息

书名：<<C#高级编程>>

13位ISBN编号：9787302184959

10位ISBN编号：730218495X

出版时间：2008-10

出版时间：内格尔 (Christian Nagel)、bill Evjen、Jay Glynn、黄静 清华大学出版社 (2008-10出版)

作者：(美)内格尔 (Nagel.C) 等著

页数：1557

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## &lt;&lt;C#高级编程&gt;&gt;

## 前言

对于开发人员来说，把C#语言及其相关环境 .NET Framework描述为多年来最重要的新技术一点都不夸张。

.NET提供了一种新环境。

在这个环境中，可以开发出运行在Windows上的几乎所有应用程序，而C#是专门用于.NET的新编程语言。

例如，使用C#可以编写出动态Web页面、XML Web服务、分布式应用程序的组件、数据库访问组件、传统的Windows桌面应用程序，甚或可以联机/脱机运行的新型智能客户应用程序。

本书介绍.NET Framework 3.5版。

如果读者使用1.0、1.1、2.0或3.0版本编码，本书的一些章节就不适用。

本书将标注出只适用于.NET Framework 3.5的新增内容。

不要被.NET这个名称所愚弄，这个名称强调Microsoft相信分布式应用程序是未来的趋势，即处理过程分布在客户机和服务器上，但C#不仅仅是编写Internet或与网络相关的应用程序的一种语言，它还提供了一种编写Windows平台上几乎任何类型的软件或组件的方式。

另外，C#和.NET都对编写程序的方式进行了革新，更易于实现在Windows上的编程。

这是一个相当重要的声明。

毕竟，我们都知道计算机技术的发展速度非常快，每年Microsoft都会推出新软件、新的编程工具或Windows的新版本，并宣称这些对开发人员都非常有用，.NET和C#也不例外。

.NET和C#的重要性为了解.NET的重要性，了解一下近10年来出现的许多Windows技术的本质，会有一定的帮助。

尽管所有的Windows操作系统在表面上看来完全不同，但从Windows 3.1（1992年）到Windows Server 2008，在内核上都有相同的Windows API。

在我们转而使用Windows的新版本时，API中增加了非常多的新功能，但这是一个演化和扩展API的过程，并非替换它。

开发Windows软件所使用的许多技术和架构也是这样。

例如，COM（Component Object Model，组件对象模型）是作为OLE（Object Linking and Embedding，对象链接和嵌入）开发出来的。

那时，它在很大程度上仅是把不同类型的Office文档链接在一起，所以利用它可以把一个小Excel电子表格放在Word文档中。

之后，它逐步演化为COM、DCOM（Distributed COM，分布式组件对象模型）和最终的COM+。COM+是一种复杂的技术，它是几乎所有组件通信方式的基础，实现了事务处理、消息传输服务和对象池。

Microsoft选择这条道路的原因非常明显：它关注向后的兼容性。

在过去的这些年中，第三方厂商编写了相当多的Windows软件，如果Microsoft每次都引入一项不遵循现有编码规则的新技术，Windows就不会获得今天的成功。

向后兼容性是Windows技术的极其重要的特性，也是Windows平台的一个长处。

但它有一个很大的缺点：每次某项技术进行演化增加了新功能后，都会比它以前更复杂。

## <<C#高级编程>>

### 内容概要

C# 经典名著！

也是Wrox皮书中最畅销的品种之一，从第1版开始就名满天下：其第3版被中华读书报、CSDN、《程序员》等机构评选为2005年最权威的十大IT图书之一；在中国版协、中国出版科学研究所、《出版参考》杂志组织的“2005年度输出版、引进版优秀图书”评选活动中获得“2005年度引进版科技类优秀图书”奖其第4版被多家IT相关机构和媒体联合评为“2006年最受读者喜爱的十大技术开发类图书”。

第6版面向C#2008和，NET 3.5，在全面展示，NET和C#新特性的同时继续完善原有的内容，是有一些C#基础知识的读者或者想迁移到C#的其他程序员全面掌握C# 2008的首选教程。

## <<C#高级编程>>

### 作者简介

作者：(美国)内格尔 (Christian Nagel) (美国)bill Evjen (美国)Jay Glynn 译者：李铭 解说词：黄静译者简介：李铭，信息管理专业硕士，拥有近8年的系统开发经验，曾开发过监考系统、宽带教学系统、人员管理系统和综合业务系统等，现就职于某部队研究所，从事.NET、Oracle和Linux等方面的开发和管理工作，曾发表过多篇计算机专业论文，也翻译和编写过几部著作。

审校简介黄静，计算机专业博士，在程序设计、软件工程和网络安全等方面有着多年的研究和实战经验，曾为.NET技术在国内的推广作出了重要贡献，翻译和发表了多篇有关.NET的技术文档或文章。

## 书籍目录

第一部分 C# 语言第1章 .NET体系结构1.1 C#与.NET的关系1.2 公共语言运行库1.2.1 平台无关性1.2.2 提高性能1.2.3 语言的互操作性1.3 中间语言1.3.1 面向对象和接口的支持1.3.2 相异值类型和引用类型1.3.3 强数据类型1.3.4 通过异常处理错误1.3.5 特性的使用1.4 程序集1.4.1 私有程序集1.4.2 共享程序集1.4.3 反射1.5 .NET Framework类1.6 命名空间1.7 用C#创建.NET应用程序1.7.1 创建ASP.NET应用程序1.7.2 创建Windows窗体1.7.3 使用Windows Presentation Foundation(WPF)1.7.4 Windows控件1.7.5 Windows服务1.7.6 Windows Communication Foundation(WCF)1.8 C#在.NET企业体系结构中的作用1.9 小结第2章 C#基础2.1 引言2.2 第一个C#程序2.2.1 代码2.2.2 编译并运行程序2.2.3 详细介绍2.3 变量2.3.1 变量的初始化2.3.2 类型推断2.3.3 变量的作用域2.3.4 常量2.4 预定义数据类型2.4.1 值类型和引用类型2.4.2 CTS类型2.4.3 预定义的值类型2.4.4 预定义的引用类型2.5 流控制2.5.1 条件语句2.5.2 循环2.5.3 跳转语句2.6 枚举2.7 数组2.8 命名空间2.8.1 using语句2.8.2 命名空间的别名2.9 Main()方法2.9.1 多个Main()方法2.9.2 给Main()方法传送参数2.10 有关编译C#文件的更多内容2.11 控制台I/O2.12 使用注释2.12.1 源文件中的内部注释2.12.2 XML文档说明2.13 C#预处理器指令2.13.1 #define和 #undef2.13.2 #if, #elif, #else和#endif2.13.3 #warning和 #error2.13.4 #region和#endregion2.13.5 #line2.13.6 #pragma2.14 C#编程规则2.14.1 用于标识符的规则2.14.2 用法约定2.15 小结第3章 对象和类型3.1 类和结构3.2 类成员3.2.1 数据成员3.2.2 函数成员3.2.3 只读字段3.3 匿名类型3.4 结构3.4.1 结构是值类型3.4.2 结构和继承3.4.3 结构的构造函数3.5 部分类3.6 静态类3.7 Object类3.7.1 System.Object方法3.7.2 ToString()方法3.8 扩展方法3.9 小结第4章 继承4.1 继承的类型4.1.1 实现继承和接口继承4.1.2 多重继承4.1.3 结构和类4.2 实现继承4.2.1 虚方法4.2.2 隐藏方法4.2.3 调用函数的基类版本4.2.4 抽象类和抽象函数4.2.5 密封类和密封方法4.2.6 派生类的构造函数4.3 修饰符4.3.1 可见性修饰符4.3.2 其他修饰符4.4 接口4.4.1 定义和实现接口4.4.2 派生的接口4.5 小结第5章 数组5.1 简单数组5.1.1 数组的声明5.1.2 数组的初始化5.1.3 访问数组元素5.1.4 使用引用类型5.2 多维数组5.3 锯齿数组5.4 Array类5.4.1 属性5.4.2 创建数组5.4.3 复制数组5.4.4 排序5.5 数组和集合接口5.5.1 IEnumerable接口5.5.2 ICollection接口5.5.3 IList接口5.6 枚举5.6.1 IEnumerator接口5.6.2 foreach语句5.6.3 yield语句5.7 小结第6章 运算符和类型强制转换6.1 运算符6.1.1 运算符的简化操作6.1.2 条件运算符6.1.3 checked和unchecked运算符6.1.4 is运算符6.1.5 as运算符6.1.6 sizeof运算符6.1.7 typeof运算符6.1.8 可空类型和运算符6.1.9 空接合运算符6.1.10 运算符的优先级6.2 类型的安全性6.2.1 类型转换6.2.2 装箱和拆箱6.3 对象的相等比较6.3.1 引用类型的相等比较6.3.2 值类型的相等比较6.4 运算符重载6.4.1 运算符的工作方式6.4.2 运算符重载的示例: Vector结构6.5 用户定义的数据类型转换6.5.1 执行用户定义的类型转换6.5.2 多重数据类型转换6.6 小结第7章 委托和事件7.1 委托7.1.1 在C#中声明委托7.1.2 在C#中使用委托7.1.3 简单的委托示例7.1.4 BubbleSorter示例7.1.5 多播委托7.1.6 匿名方法7.1.7 ?表达式7.1.8 协变和抗变7.2 事件7.2.1 从接收器的角度讨论事件7.2.2 生成事件7.3 小结第8章 字符串和正则表达式8.1 System.String类8.1.1 创建字符串8.1.2 StringBuilder成员8.1.3 格式化字符串8.2 正则表达式8.2.1 正则表达式概述8.2.2 RegularExpressionsPlayaround示例8.2.3 显示结果8.2.4 匹配、组合和捕获8.3 小结第9章 泛型9.1 概述9.1.1 性能9.1.2 类型安全9.1.3 二进制代码的重用9.1.4 代码的扩展9.1.5 命名约定9.2 创建泛型类9.3 泛型类的特性9.3.1 默认值9.3.2 约束9.3.3 继承9.3.4 静态成员9.4 泛型接口9.5 泛型方法9.6 泛型委托9.6.1 执行委托调用的方法9.6.2 对Array类使用泛型委托9.7 Framework的其他泛型类型9.7.1 结构Nullable9.7.2 EventHandler9.7.3 ArraySegment9.8 小结第10章 集合10.1 集合接口和类型10.2 列表10.2.1 创建列表10.2.2 只读集合10.3 队列10.4 栈10.5 链表10.6 有序表10.7 字典10.7.1 键的类型10.7.2 字典示例10.7.3 Lookup类10.7.4 其他字典类10.8 HashSet10.9 位数组10.9.1 BitArray10.9.2 BitVector3210.10 性能10.11 小结第11章 Language Integrated Query11.1 LINQ概述11.1.1 使用List的查询11.1.2 扩展方法11.1.3 表达式11.1.4 LINQ查询11.1.5 推迟查询的执行11.2 标准的查询操作符11.2.1 过滤11.2.2 用索引来过滤11.2.3 类型过滤11.2.4 复合的from子句11.2.5 排序11.2.6 分组11.2.7 对嵌套的对象分组11.2.8 连接11.2.9 设置操作11.2.10 分区11.2.11 合计操作符11.2.12 转换11.2.13 生成操作符11.3 表达式树11.4 LINQ提供程序11.5 小结第12章 内存管理和指针12.1 后台内存管理12.1.1 值数据类型12.1.2 引用数据类型12.1.3 垃圾收集12.2 释放未托管的资源12.2.1 析构函数12.2.2 IDisposable接口12.2.3 实现IDisposable接口和析构函数12.3 不安全的代码12.3.1 用指针直接访问内存12.3.2 指针示例: PointerPlayaround12.3.3 使用指针优化性能12.4 小结第13章 反射13.1 定制特

性13.1.1 编写定制特性13.1.2 定制特性示例：WhatsNewAttributes13.2 反射13.2.1 System.Type类13.2.2  
TypeView示例13.2.3 Assembly类13.2.4 完成WhatsNewAttributes示例13.3 小结第14章 错误和异常14.1 异常  
类14.2 捕获异常14.2.1 执行多个catch块14.2.2 在其他代码中捕获异常14.2.3 System.Exception属性14.2.4 没  
有处理异常时所发生的情况14.2.5 嵌套的 try块14.3 用户定义的异常类14.3.1 捕获用户定义的异常14.3.2  
抛出用户定义的异常14.3.3 定义异常类14.4 小结第 一部分 Visual Studio第15章 Visual Studio 200815.1 使  
用Visual Studio 200815.1.1 创建项目15.1.2 解决方案和项目15.1.3 Windows应用程序代码15.1.4 读取Visual  
Studio 6项目15.1.5 项目的浏览和编码15.1.6 生成项目15.1.7 调试15.2 修订功能15.3 多目标15.4 WPF  
、WCF、WF等15.4.1 在Visual Studio中建立WPF应用程序15.4.2 在Visual Studio中建立WF应用程序15.5  
小结第16章 部署16.1 部署的设计16.2 部署选项16.2.1 Xcopy实用工具16.2.2 Copy Web工具16.2.3 发布Web  
站点16.2.4 部署项目16.2.5 ClickOnce16.3 部署的要求16.4 部署.NET运行库16.5 简单的部署16.5.1 Xcopy部  
署16.5.2 Xcopy和Web应用程序16.5.3 Copy Web工具16.5.4 发布Web站点16.6 Installer项目16.6.1 Windows  
Installer16.6.2 创建安装程序16.7 ClickOnce16.7.1 ClickOnce操作16.7.2 发布应用程序16.7.3 ClickOnce设  
置16.7.4 应用程序缓存16.7.5 安全性16.7.6 高级选项16.8 小结第 一部分 基类库第17章 程序集17.1 程序集  
的含义17.1.1 程序集的特性17.1.2 程序集的结构17.1.3 程序集的清单17.1.4 命名空间、程序集和组  
件17.1.5 私有程序集和共享程序集17.1.6 辅助程序集17.1.7 查看程序集17.2 构建程序集17.2.1 创建模块和  
程序集17.2.2 程序集的属性17.3 动态加载和创建程序集17.4 应用程序域17.5 共享程序集17.5.1 强名17.5.2  
使用强名获得完整性17.5.3 全局程序集缓存17.5.4 创建共享程序集17.5.5 创建强名17.5.6 安装共享程序  
集17.5.7 使用共享程序集17.5.8 程序集的延迟签名17.5.9 引用17.5.10 本机图像生成器17.6 配置.NET应用  
程序17.6.1 配置类别17.6.2 为搜索程序集配置目录17.7 版本问题17.7.1 版本号17.7.2 编程获取版本17.7.3  
应用程序配置文件17.7.4 发布方的策略文件17.7.5 运行库的版本17.8 小结第18章 跟踪和事件18.1 跟  
踪18.1.1 跟踪源18.1.2 跟踪开关18.1.3 跟踪监听器18.1.4 过滤器18.1.5 断言18.2 事件日志18.2.1 事件日志体  
系架构18.2.2 事件日志类18.2.3 创建事件源18.2.4 写入事件日志18.2.5 资源文件18.3 性能监控18.3.1 性能  
监控类18.3.2 性能计数器的构建18.3.3 添加PerformanceCounter组件18.3.4 perfmon.exe18.4 小结第19章 线  
程和同步19.1 概述19.2 异步委托19.2.1 投票19.2.2 等待句柄19.2.3 异步回调19.3 Thread类19.3.1 给线程传  
送数据19.3.2 后台线程19.3.3 线程的优先级19.3.4 控制线程19.4 线程池19.5 线程问题19.5.1 竞态条件19.5.2  
死锁19.6 同步19.6.1 lock语句和线程安全19.6.2 Interlocked19.6.3 Monitor类19.6.4 等待句柄19.6.5 Mutex  
类19.6.6 Semaphore类19.6.7 Events类19.6.8 ReaderWriterLockSlim19.7 Timer类19.8 COM单元19.9 基于事件  
的异步模式19.9.1 BackgroundWorker类19.9.2 激活取消功能19.9.3 激活进度功能19.9.4 创建基于事件的异  
步组件19.10 小结第20章 安全性20.1 身份验证和授权20.1.1 标识和Principal20.1.2 角色20.1.3 声明基于角色  
的安全性20.1.4 客户应用程序服务20.2 加密20.2.1 签名20.2.2 密钥的互换和安全传送20.3 资源的访问控  
制20.4 代码访问的安全性20.4.1 权限20.4.2 代码组20.4.3 策略的级别：Machine、User和Enterprise20.5 安  
全策略的管理20.5.1 代码组和权限的管理20.5.2 安全性的启用和禁用20.5.3 代码组的创建20.5.4 代码组的  
删除20.5.5 代码组权限的更改20.5.6 权限集的创建和应用20.5.7 使用强名发布代码20.5.8 使用证书发布代  
码20.6 小结第21章 本地化21.1 System.Globalization命名空间21.1.1 Unicode问题21.1.2 文化和区域21.1.3 使  
用文化21.1.4 排序21.2 资源21.2.1 创建资源文件21.2.2 资源文件生成器21.2.3 ResourceWriter21.2.4 使用资  
源文件21.2.5 System.Resources命名空间21.3 使用Visual Studio的Windows Forms本地化示例21.3.1 编程修  
改文化21.3.2 使用定制资源文件21.3.3 资源的自动回退21.3.4 外包翻译21.4 用ASP.NET本地化21.5 用WPF  
本地化21.5.1 WPF应用程序21.5.2 .NET资源21.5.3 用XAML本地化21.6 定制的资源读取器21.6.1 创  
建DatabaseResourceReader类21.6.2 创建DatabaseResourceSet类21.6.3 创建DatabaseResource-Manager  
类21.6.4 DatabaseResourceReader的客户应用程序21.7 创建定制文化21.8 小结第22章 事务处理22.1 概  
述22.1.1 事务处理阶段22.1.2 ACID属性22.2 数据库和实体类22.3 传统的事务处理22.3.1 ADO.NET事务处  
理22.3.2 System.EnterpriseServices22.4 System.Transactions22.4.1 可提交的事务处理22.4.2 事务处理的升  
级22.4.3 依赖的事务处理22.4.4 环境事务处理22.5 隔离级别22.6 定制资源管理器22.7 Windows Vista  
和Windows Server 2008 的事务处理22.8 小结第23章 Windows服务23.1 Windows服务23.2 Windows服务的  
体系结构23.2.1 服务程序23.2.2 服务控制程序23.2.3 服务配置程序23.3 System.ServiceProcess命名空间23.4  
创建Windows服务23.4.1 使用套接字的类库23.4.2 TcpClient示例23.4.3 Windows服务项目23.4.4 线程和服  
务23.4.5 服务的安装23.4.6 安装程序23.5 服务的监视和控制23.5.1 MMC计算机管理23.5.2 net.exe23.5.3

sc.exe 23.5.4 Visual Studio Server Explorer 23.5.5 ServiceController类 23.6 故障排除 23.6.1 交互式服务 23.6.2 事件日志 23.7 电源事件 23.8 小结第24章 互操作性 24.1 .NET和COM 24.1.1 元数据 24.1.2 释放内存 24.1.3 接口 24.1.4 方法绑定 24.1.5 数据类型 24.1.6 注册 24.1.7 线程 24.1.8 错误处理 24.1.9 事件处理 24.2 编组 24.3 在.NET客户程序中使用COM组件 24.3.1 创建COM组件 24.3.2 创建Runtime Callable Wrapper 24.3.3 使用RCW 24.3.4 可交互操作的主程序集 24.3.5 线程问题 24.3.6 添加连接点 24.3.7 在Windows窗体中使用ActiveX控件 24.3.8 在ASP.NET中使用COM对象 24.4 在COM客户程序中使用.NET组件 24.4.1 COM Callable Wrapper 24.4.2 创建.NET组件 24.4.3 创建类型库 24.4.4 COM互操作特性 24.4.5 COM注册 24.4.6 创建COM客户程序 24.4.7 添加连接点 24.4.8 用sink对象创建客户程序 24.4.9 在Internet Explorer中运行Windows窗体控件 24.5 平台调用 24.6 小结第 一部分 数据第25章 文件和注册表操作 25.1 管理文件系统 25.1.1 表示文件和文件夹的.NET类 25.1.2 Path类 25.1.3 示例：文件浏览器 25.2 移动、复制和删除文件 25.2.1 FilePropertiesAndMovement示例 25.2.2 示例FilePropertiesAnd-Movement的代码 25.3 读写文件 25.3.1 读取文件 25.3.2 写入文件 25.3.3 流 25.3.4 缓存的流 25.3.5 使用FileStream类读写二进制文件 25.3.6 读写文本文件 25.4 读取驱动器信息 25.5 文件的安全性 25.5.1 从文件中读取ACL 25.5.2 从目录中读取ACL 25.5.3 添加和删除文件中的ACL项 25.6 读写注册表 25.6.1 注册表 25.6.2 .NET注册表类 25.6.3 SelfPlacingWindow示例 25.7 读写独立存储器 25.8 小结第26章 .NET数据访问 26.1 ADO.NET概述 26.1.1 命名空间 26.1.2 共享类 26.1.3 数据库特定的类 26.2 使用数据库连接 26.2.1 管理连接字符串 26.2.2 高效地使用连接 26.2.3 事务处理 26.3 命令 26.3.1 执行命令 26.3.2 调用存储过程 26.4 快速数据访问：数据读取器 26.5 管理数据和关系：DataSet类 26.5.1 数据表 26.5.2 数据列 26.5.3 数据关系 26.5.4 数据约束 26.6 XML模式：用XSD生成代码 26.7 填充数据集 26.7.1 用数据适配器来填充DataSet 26.7.2 从XML中给数据集填充数据 26.8 保存对数据集的修改 26.8.1 通过数据适配器进行更新 26.8.2 写入XML输出结果 26.9 使用ADO.NET 26.9.1 分层开发 26.9.2 生成SQL Server的键 26.9.3 命名约定 26.10 小结第27章 LINQ to SQL 27.1 LINQ to SQL和Visual Studio 2008 27.1.1 使用LINQ to SQL调用Products表——创建控制台应用程序 27.1.2 添加LINQ to SQL类 27.1.3 O/R设计器概述 27.1.4 创建Product对象 27.2 对象映射到LINQ对象上 27.2.1 DataContext对象 27.2.2 Table对象 27.3 不使用O/R设计器工作 27.3.1 创建自己的定制对象 27.3.2 用定制的对象和LINQ进行查询 27.3.3 用查询限制所调用的列 27.3.4 使用列名 27.3.5 创建自己的DataContext对象 27.4 定制对象和O/R设计器 27.5 查询数据库 27.5.1 使用查询表达式 27.5.2 查询表达式 27.5.3 使用表达式过滤 27.5.4 连接 27.5.5 组合数据项 27.6 存储过程 27.7 小结第28章 处理XML 28.1 .NET支持的XML标准 28.2 System.Xml命名空间 28.3 使用System.Xml类 28.4 读写流格式的XML 28.4.1 使用 XmlReader类 28.4.2 使用XmlReader类进行验证 28.4.3 使用XmlWriter类 28.5 在.NET中使用DOM 28.6 使用XPathNavigator 28.6.1 System.Xml.XPath命名空间 28.6.2 System.Xml.Xsl命名空间 28.6.3 调试XSLT 28.7 XML和ADO.NET 28.7.1 将ADO.NET数据转换为XML文档 28.7.2 把XML文档转换为ADO.NET数据 28.8 在XML中串行化对象 28.9 小结第29章 LINQ to XML 29.1 LINQ to XML和.NET 3.5 29.1.1 创建XML文档的新对象 29.1.2 Visual Basic 2008开辟了另一条道路 29.1.3 命名空间和前缀 29.2 .NET Framework 3.5中的新XML对象 29.2.1 XDocument对象 29.2.2 XElement对象 29.2.3 XNamespace对象 29.2.4 XComment对象 29.2.5 XAttribute对象 29.3 使用LINQ查询XML文档 29.3.1 查询静态的XML文档 29.3.2 查询动态的XML文档 29.4 处理XML文档 29.4.1 读取XML文档 29.4.2 写入XML文档 29.5 使用LINQ to SQL和LINQ to XML 29.5.1 建立LINQ to SQL组件 29.5.2 查询数据库，输出XML 29.6 小结第30章 .NET编程和SQL Server 30.1 .NET运行库的主机 30.2 Microsoft.SqlServer.Server 30.3 用户定义的类型 30.3.1 创建UDT 30.3.2 使用UDT 30.3.3 在客户端代码中使用UDT 30.4 用户定义的合计函数 30.4.1 创建用户定义的合计函数 30.4.2 使用用户定义的合计函数 30.5 存储过程 30.5.1 创建存储过程 30.5.2 使用存储过程 30.6 用户定义的函数 30.6.1 创建用户定义的函数 30.6.2 使用用户定义的函数 30.7 触发器 30.7.1 创建触发器 30.7.2 使用触发器 30.8 XML数据类型 30.8.1 包含XML数据的表 30.8.2 读取XML值 30.8.3 数据的查询 30.8.4 XML数据修改语言(XML DML) 30.8.5 XML索引 30.8.6 强类型化的XML 30.9 小结第 一部分 显示第31章 Windows窗体 31.1 创建Windows窗体应用程序类层次结构 31.2 Control类 31.2.1 大小和位置 31.2.2 外观 31.2.3 用户交互操作 31.2.4 Windows功能 31.2.5 杂项功能 31.3 标准控件和组件 31.3.1 Button控件 31.3.2 CheckBox控件 31.3.3 RadioButton控件 31.3.4 ComboBox控件、ListBox控件和CheckedListBox控件 31.3.5 DateTimePicker控件 31.3.6 ErrorProvider组件 31.3.7 HelpProvider组件 31.3.8 ImageList组件 31.3.9 Label控件 31.3.10 ListView控件 31.3.11 PictureBox控件 31.3.12 ProgressBar控件 31.3.13 TextBox控件、RichTextBox控件与MaskedTextBox

控件31.3.14 Panel控件31.3.15 FlowLayoutPanel和TableLayoutPanel控件31.3.16 SplitContainer控件31.3.17 TabControl控件和TabPage控件31.3.18 ToolStrip控件31.3.19 MenuStrip控件31.3.20 ContextMenuStrip控件31.3.21 ToolStripMenuItem控件31.3.22 ToolStripManager类31.3.23 ToolStripContainer控件31.4 窗体31.4.1 Form类31.4.2 多文档界面31.4.3 定制控件31.5 小结第32章 数据绑定32.1 DataGridView控件32.1.1 显示列表数据32.1.2 数据源32.2 DataGridView类的层次结构32.3 数据绑定32.3.1 简单的绑定32.3.2 数据绑定对象32.4 Visual Studio和数据访问32.4.1 创建一个连接32.4.2 选择数据32.4.3 更新数据源32.4.4 其他常见的要求32.5 小结第33章 使用GDI+绘图33.1 理解绘图规则33.1.1 GDI和GDI+33.1.2 绘制图形33.1.3 使用OnPaint()绘制图形33.1.4 使用剪切区域33.2 测量坐标和区域33.2.1 Point和PointF结构33.2.2 Size和SizeF结构33.2.3 Rectangle和RectangleF结构33.2.4 Region33.3 调试须知33.4 绘制可滚动的窗口33.5 世界、页面和设备坐标33.6 颜色33.6.1 红绿蓝(RGB)值33.6.2 命名的颜色33.6.3 图形显示模式和安全的调色板33.6.4 安全调色板33.7 画笔和钢笔33.7.1 画笔33.7.2 钢笔33.8 绘制图形和线条33.9 显示图像33.10 处理图像时的问题33.11 绘制文本33.12 简单的文本示例33.13 字体和字体系列33.14 示例：枚举字体系列33.15 编辑本文档：CapsEditor示例33.15.1 Invalidate()方法33.15.2 计算项和文档的大小33.15.3 OnPaint()33.15.4 坐标转换33.15.5 响应用户的输入33.16 打印33.17 小结第34章 Windows Presentation Foundation34.1 概述34.1.1 XAML34.1.2 设计人员和开发人员的合作34.1.3 类层次结构34.1.4 命名空间34.2 形状34.3 变换34.4 笔刷34.4.1 SolidColorBrush34.4.2 LinearGradientBrush34.4.3 RadialGradientBrush34.4.4 DrawingBrush34.4.5 ImageBrush34.4.6 VisualBrush34.5 控件34.5.1 简单控件34.5.2 内容控件34.5.3 有标题的内容控件34.5.4 项控件34.5.5 带标题的项控件34.6 布局34.6.1 StackPanel34.6.2 WrapPanel34.6.3 Canvas34.6.4 DockPanel34.6.5 Grid34.7 事件处理34.8 样式、模板和资源34.8.1 样式34.8.2 资源34.8.3 创建列表框的样式34.9 小结第35章 高级WPF35.1 数据绑定35.1.1 概述35.1.2 用XAML绑定35.1.3 简单对象的绑定35.1.4 对象数据提供程序35.1.5 列表绑定35.1.6 绑定到XML上35.1.7 绑定的验证35.2 命令绑定35.3 动画35.3.1 时间线35.3.2 触发器35.3.3 故事板35.4 在WPF中添加3D特性35.5 Windows窗体集成35.5.1 Windows窗体中的WPF控件35.5.2 WPF应用程序中的Windows窗体控件35.6 WPF浏览器应用程序35.7 小结第36章 插件36.1 System.AddIn体系结构36.1.1 插件的问题36.1.2 管道体系结构36.1.3 发现36.1.4 激活和隔离36.1.5 合同36.1.6 生存期36.1.7 版本问题36.2 插件示例36.2.1 计算器合同36.2.2 计算器插件视图36.2.3 计算器插件适配器36.2.4 计算器插件36.2.5 计算器主机视图36.2.6 计算机主机适配器36.2.7 计算器主机36.2.8 其他插件36.3 小结第37章 ASP.NET页面37.1 ASP.NET概述37.2 ASP.NET Web窗体37.2.1 ASP.NET代码模型37.2.2 ASP.NET服务器控件37.3 ADO.NET和数据绑定37.3.1 更新会议登记应用程序37.3.2 数据绑定的更多内容37.4 应用程序配置37.5 小结第38章 ASP.NET开发38.1 用户控件和定制控件38.1.1 用户控件38.1.2 PCSDemoSite中的用户控件38.1.3 定制控件38.2 Master页面38.2.1 在Web页面中访问Master页面38.2.2 嵌套的Master页面38.2.3 PCSDemoSite中的Master页面38.3 站点导航38.4 安全性38.4.1 使用安全向导添加Forms身份验证功能38.4.2 实现登录系统38.4.3 Web 登录服务器控件38.4.4 保护目录38.4.5 PCSDemoSite中的安全性38.5 主题38.5.1 把主题应用于页面38.5.2 定义主题38.5.3 PCSDemoSite中的主题38.6 Web Parts38.6.1 Web Parts应用程序组件38.6.2 Web Parts示例38.7 小结第39章 ASP.NET AJAX39.1 Ajax的概念39.2 ASP.NET AJAX39.2.1 核心功能39.2.2 ASP.NET AJAX Control Toolkit39.3 使用ASP.NET AJAX39.3.1 ASP.NET AJAX网站示例39.3.2 支持ASP.NET AJAX的网站配置39.3.3 添加ASP.NET AJAX功能39.3.4 使用AJAX库39.4 小结第40章 Visual Studio Tools for Office40.1 VSTO概述40.1.1 项目类型40.1.2 项目特性40.2 VSTO基础40.2.1 Office对象模型40.2.2 VSTO命名空间40.2.3 主机项和主机控件40.2.4 基本的VSTO项目结构40.2.5 Globals类40.2.6 事件处理40.3 建立VSTO解决方案40.3.1 管理应用程序级插件40.3.2 与应用程序和文档交互操作40.3.3 UI的定制40.4 示例应用程序40.5 VBA交互操作性40.6 小结第 部分 通信第41章 访问Internet41.1 WebClient类41.1.1 下载文件41.1.2 基本的Web客户示例41.1.3 上传文件41.2 WebRequest类和WebResponse类41.3 把输出结果显示为HTML页面41.3.1 在应用程序中进行简单的Web浏览41.3.2 启动Internet Explorer实例41.3.3 给应用程序提供更多的IE类型特性41.3.4 使用WebBrowser控件打印41.3.5 显示请求页面的代码41.3.6 WebRequest和WebResponse的层次结构41.4 实用工具类41.4.1 URI41.4.2 IP地址和DNS名称41.5 较低层的协议41.6 小结第42章 Windows Communication Foundation42.1 WCF概述42.1.1 SOAP42.1.2 WSDL42.1.3 JSON42.2 简单的服务和客户42.2.1 服务合同42.2.2 服务的实现42.2.3 WCF服务主机和WCF测试客户机42.2.4 定制服务主机42.2.5 WCF客户程序42.2.6 诊断42.3 合同42.3.1 数据合同42.3.2 版本问



题42.3.3 服务合同42.3.4 消息合同42.4 服务的实现42.5绑定42.6 主机42.6.1 定制主机42.6.2 WAS主机42.7 客户程序42.8 双向通信42.9 小结第43章 Windows Workflow Foundation43.1 Hello World示例43.2 活动43.2.1 IfElseActivity43.2.2 ParallelActivity43.2.3 CallExternalMethodActivity43.2.4 DelayActivity41.2.5 ListenActivity43.2.6 活动的执行模型43.3 定制的活动43.3.1 活动的有效性验证43.3.2 主题和设计器43.3.3 ActivityToolboxItem和图标43.3.4 定制的复合活动43.4 工作流43.4.1 系列工作流43.4.2 状态机工作流43.4.3 给工作流传送参数43.4.4 从工作流中返回结果43.4.5 将参数绑定到活动上43.5 工作流运行库43.6 工作流服务43.6.1 持续服务43.6.2 跟踪服务43.6.3 定制服务43.7 与WCF集成43.8 保存工作流43.9 工作流设计器43.10 小结第44章 Enterprise Services44.1 概述44.1.1 Enterprise Services简史44.1.2 使用Enterprise Services的场合44.1.3 环境44.1.4 自动的事务处理44.1.5 分布式事务处理44.1.6 对象池44.1.7 基于角色的安全性44.1.8 排队的组件44.1.9 松散耦合的事件44.2 创建简单的COM+应用程序44.2.1 类ServicedComponent44.2.2 标记程序集44.2.3 程序集的属性44.2.4 创建组件44.3 部署44.3.1 自动部署44.3.2 手工部署44.3.3 创建安装软件包44.4 Component Services浏览器44.5 客户应用程序44.6 事务处理44.6.1 事务处理的属性44.6.2 事务处理的结果44.7 示例应用程序44.7.1 实体类44.7.2 OrderControl组件44.7.3 OrderData组件44.7.4 OrderLineData组件44.7.5 客户应用程序44.8 集成WCF和Enterprise Services44.8.1 WCF服务Fa?ade44.8.2 客户程序44.9 小结第45章 消息队列45.1 概述45.1.1 使用Message Queuing的场合45.1.2 Message Queuing特性45.2 Message Queuing产品45.3 Message Queuing结构45.3.1 消息45.3.2 消息队列45.4 Message Queuing管理工具45.4.1 创建消息队列45.4.2 消息队列属性45.5 Message Queuing的编程实现45.5.1 创建消息队列45.5.2 查找队列45.5.3 打开已知的队列45.5.4 发送消息45.5.5 接收消息45.6 课程订单应用程序45.6.1 课程订单类库45.6.2 课程订单消息发送程序45.6.3 发送优先级和可恢复的消息45.6.4 课程订单消息接收程序45.7 接收结果45.7.1 确认队列45.7.2 响应队列45.8 事务队列45.9 消息队列和WCF45.9.1 带数据合同的实体类45.9.2 WCF服务合同45.9.3 WCF消息接收程序45.9.4 WCF消息发送程序45.10 消息队列的安装45.11 小结第46章 目录服务46.1 Active Directory的体系结构46.1.1 特性46.1.2 Active Directory的概念46.1.3 Active Directory数据的特性46.1.4 模式46.2 Active Directory的管理工具46.2.1 Active Directory Users and Computers工具46.2.2 ADSI Edit工具46.3 Active Directory编程46.3.1 System.DirectoryServices命名空间中的类46.3.2 绑定46.3.3 获取目录项46.3.4 对象集合46.3.5 缓存46.3.6 创建新对象46.3.7 更新目录项46.3.8 访问内部的ADSI对象46.3.9 在Active Directory中搜索46.4 搜索用户对象46.4.1 用户界面46.4.2 获取模式命名环境46.4.3 获取User类的属性名46.4.4 搜索用户对象46.5 账户管理46.5.1 显示用户信息46.5.2 创建用户46.5.3 重置密码46.5.4 创建组46.5.5 在组中添加用户46.5.6 查找用户46.6 DSML46.6.1 命名空间System.Directory-Services.Protocols中的类46.6.2 用DSML搜索Active Directory对象46.7 小结第47章 对等网络47.1 P2P概述47.1.1 客户机-服务器体系结构47.1.2 P2P体系结构47.1.3 P2P体系结构的挑战47.1.4 P2P术语47.1.5 P2P解决方案47.2 Microsoft Windows Peer-to-Peer Networking47.2.1 Peer Name Resolution Protocol(PNRP)47.2.2 People Near Me47.3 建立P2P应用程序47.3.1 System.Net.PeerToPeer47.3.2 System.Net.PeerToPeer.Collaboration47.4 小结第48章 Syndication48.1 System.ServiceModel.Syndication命名空间概述48.2 Syndication阅读器48.3 提供SyndicationFeed48.4 小结第 部分 附录附录A ADO.NET Entity Framework附录B C#、Visual Basic和C++/CLI附录C Windows Vista和Windows Server 2008

章节摘录

插图：

## <<C#高级编程>>

### 编辑推荐

《C#高级编程(第6版)》是C#经典名著！  
也是Wrox皮书中最畅销的品种之一，从第1版开始就名满天下：其第3版被中华读书报、CSDN、《程序员》等机构评选为2005年最权威的十大IT图书之一；在中国版协、中国出版科学研究所、《出版参考》杂志组织的“2005年度输出版、引进版优秀图书”评选活动中获得“2005年度引进版科技类优秀图书”奖其第4版被多家IT相关机构和媒体联合评为“2006年最受读者喜爱的十大技术开发类图书”。  
Wrox 畅销书！  
C#经典名著2005年最权威的十大IT图书！  
2005年度引进版科技类优秀图书！  
2006年最爱读者喜爱的十大技术开发类图书！  
2007年最畅销的C#零售图书！  
之C#2008最新版。

#### 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>