

Broadview®
www.broadview.com.cn

基础 ▶ 案例 ▶ 实践
一本全面讲解C#的学习宝典!

完全手册

Visual C# 2008

开发技术详解



程序源代码+多媒体入门视频+
5个完整典型系统+659页相关技术实用学习电子书

李容 等编著

本书=全面丰富的内容+统一规范的讲解+详细超值的光盘+完善及时的答疑



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn



完全手册

Visual C# 2008 开发技术详解

李容 等编著

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING



内 容 简 介

C#是 Microsoft 专门为基于.NET Framework 平台开发而量身定做的编程语言。本书全面细致地介绍了 Visual Studio 2008 环境下的 C#编程技术。书中每个知识和技术要点都给予了翔实的例程及代码分析。

本书从 C#入门基础知识到 C#的基本语法、从面向对象基础到泛型程序设计、从数据库操作到网络与安全编程、从 ADO.NET 到 XML 操作、从控制台应用程序开发到 ASP.NET Web 应用程序开发,再到 Web 服务程序开发等内容作了详细介绍,几乎涵盖了 C#相关的所有开发技术。同时以较大篇幅介绍了 C# 2008 中的重要特性,如泛型、GDI+、ASP.NET,以及数据操作控件等。最后介绍两个案例,并给出详细的设计过程及代码说明,以巩固和加深本书介绍的知识。

全书配合了大量的图例及代码说明,如果你希望掌握有关 C#面向对象编程技术,那么本书将是不错的选择;如果你已经有这方面的编程经验,本书也可以作为学习和开发的一个很好的参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

完全手册 Visual C# 2008 开发技术详解 / 李容等编著. —北京:电子工业出版社,2008.5

(完全手册)

ISBN 978-7-121-06318-3

I. 完… II. 李… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 044064 号

责任编辑:高洪霞

印 刷:北京天宇星印刷厂

装 订:三河市皇庄路通装订厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:46.75 字数:1149 千字

印 次:2008 年 5 月第 1 次印刷

印 数:5000 册 定价:79.80 元(含光盘 1 张)

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前言

随着.NET 3.5 平台的不断升级和完善,越来越多的企业、单位和个人开始关注并基于该平台进行软件开发。鉴于该平台具有完善而优美的表现力,且又不乏实现的简洁性和平滑的学习曲线,足以使得每个使用它的程序员爱不释手。

本书以教程的形式深入浅出、由易到难详细地介绍了.NET 平台下,关于C#的所有开发技术。每个知识和技术要点都以例题讲解为主,同时辅以必要的原理和理论说明。这也是程序员特别是初学者非常欢迎的讲解方式。

本书以在展示C#基本编程技能的同时,让读者提高解决实际问题的能力为主旨。通过对大量经典例程的说明,将基础知识与实际应用巧妙结合,使读者在学习过程中既掌握了基本知识,又提高了解决实际问题的灵活性。书中很多代码只需稍加修改,便可以应用于实际开发中。

需要说明的是,由于本书编写完成时正值微软推出 Visual Studio 2008 正式版。鉴于此,考虑到部分读者朋友已经或者打算使用 Visual Studio 2008,编者特地进行了版本升级。将书中基于 Visual Studio 2005 编写的程序移植到 Visual Studio 2008 平台。由于这两个平台中基于.NET 2.0 框架部分的程序开发是一致的,同时它们都支持.NET 2.0,因此本书既适用于 Visual Studio 2008 程序开发,也适用于 Visual Studio 2005 程序开发。

本书的内容安排

本书全面细致地介绍了 Visual Studio 2008 环境下的C#编程技术。书中每个知识和技术要点都给予了详实的例程及代码分析。全书共分七篇,内容涵盖了C#入门知识、C#语法及面向对象基础、窗体应用程序开发、C#数据存取、网络和安全编程、Web 编程以及综合案例等内容。

第一篇(第1章)是关于C#入门的基础知识,主要对Microsoft.NET 框架体系结构、C#语言的特点及其主要开发平台 Visual Studio 2008 作了详细的介绍。最后,介绍了利用C#语言在 Visual Studio 2008 上开发应用程序的基本方法,以及其涉及的操作技巧。这些程序包括 Windows 控制台应用程序、Windows 窗体应用程序和 ASP.NET 应用程序。

第二篇(第2章~第7章)介绍了C#基本语法与操作,包括C#语法基础、C#面向对象基础、字符串处理、数组处理、集合与泛型、代码调试等内容。这部分内容是C#应用程序开发的基础,也是后续内容的预备知识。

第三篇(第8章~第12章)是有关 Windows 窗体应用程序开发的知识。主要内容有编写 Windows 窗体应用程序、使用 Windows 窗体控件、GDI+简介、文件操作、部署 Windows 应用程序。

第四篇(第13章~第15章)介绍了在C#中操作数据的基本知识。这部分主要包括了数据库基础应用知识、利用 ADO.NET 访问数据库、在.NET 中访问 XML 等内容。

第五篇(第16章~第19章)是有关网络和安全编程。其中详细介绍了线程、Socket

套接字编程、.NET 安全编程、.NET 中的电子邮件处理有关内容。

第六篇（第 20 章～第 24 章）介绍了有关基于 ASP.NET 的 Web 应用程序开发方面的内容。其中包括 Web 编程基础、Web 站点的构建、Web 程序中服务器端的交互控件、ASP.NET 3.5 高级编程、C#操作 Web Service 等内容。

第七篇（第 25 章～第 26 章）是两个完整的案例，它们是图书管理系统和学生信息管理系统。其中给出了详细的设计过程及代码说明，以巩固和加深本书所介绍的知识点。

本书全面而系统地介绍了基于 .NET 3.5 的 C#编程。书中的每个知识和技术要点都配合了经典的实例讲解，每段代码都给出了注释与分析，旨在使读者朋友能够透彻理解。

本书的特点

1. 配有学习论坛

论坛网址是 <http://www.rzchina.net>，读者可以在上面讨论技术，笔者会及时回答读者提问，并提供各种技术文章，帮助你提高开发水平。

2. 全面而丰富的内容

本书分 7 篇，共 26 章，采用从易到难、循序渐进的方式进行讲解。内容几乎涉及了 C#程序设计的各个方面。不但如此，书中还增加了程序设计相关重要领域的部分知识，如 Web 应用程序开发中的脚本设计、SEO 等内容。

3. 统一而规范的例程讲解方式

本书每个例程都采用了统一的分步骤实现方法，这样使得读者可以很清晰地知道每个技术的具体实现步骤，从而提高学习的效率。

4. 较高的实用价值

本书给出了大量的经典实例，这些实例都已在 .NET 平台上调试成功。同时，书中给出了代码的详细注释与分析，读者只需对某些代码稍加修改，便可以将其应用于实际开发中。

5. 清晰透彻的讲解

本书主要为初、中级程序开发人员编写，因此在知识点的讲解过程中尽量做到通俗易懂，简洁明了。在保证阐述严谨的同时，力求做到容易理解，不钻牛角尖，不使用过于专业、晦涩艰深的术语，不使用有歧义的表达方式，尽量使读者在阅读时能够迅速掌握关键知识点。

本书的学习流程

不同阶段的学习人员，可从下页图中概览本书的组织结构，找到自己的学习位置，更好、更快地学习本书中的知识。

适合阅读本书的读者

- 使用 Visual Studio 2005/2008 的程序设计人员

- 已经有.NET 平台开发经验，但又想转到.NET 2.0、.NET 3.0、.NET 3.5 平台开发的技术人员
- 正在使用.NET 平台开发的工作人员
- 打算全面了解并学习基于.NET 3.5 程序开发的人员
- 使用 SQL Server 2005 数据库进行网站开发的人员
- 从其他语言转行学习 C#语言的人员



致谢

特别感谢余勇副教授和王贤勤研究员，在本书的编写过程中他们给予了很多有建设性的建议和帮助。参与本书编写以及资料整理的人员还有黄仲、武娜、王康、田泽彬、吕博、唐勇、纪颖，感谢他们的辛勤劳动。

虽然本书经过了多次反复的修改，力求展现出当前计算机程序设计的先进技术。但是计算机科学技术发展迅速，加之我们水平有限、时间仓促，书中仍难免会有不少缺点或错漏，恳请读者批评指正。

编 者

2008 年 3 月

目 录

Contents

第一篇 C# 2008 入门必备

第 1 章 认识 C#

2

本章首先介绍 C#语言相关的基本知识,以使读者对 C#有一个整体的认识。

C#是目前世界上最好的编程语言之一,其集中了几乎所有关于软件开发和软件工程研究领域的最新成果。C#拥有完全面向对象、类型安全、组件技术、版本控制、自动内存管理、跨平台异常处理,以及代码安全管理等先进特性。这些特性使它成为众多程序员和企业的首选开发语言。

1.1 什么是 C#	2
1.1.1 Microsoft .NET 框架	2
1.1.2 .NET 框架的开发语言——C#	3
1.1.3 C#的开发平台 Visual Studio 2008	3
1.2 第一个 C#程序	4
1.2.1 认识和配置 Visual Studio 2008 开发环境	4
1.2.2 第一个 Windows Console 程序	5
1.2.3 创建 Windows Console 程序	5
1.2.4 解决方案资源管理器	7
1.2.5 属性窗口	7
1.2.6 第一个 Windows Form 程序	8
1.2.7 创建 Windows Form 程序	8
1.2.8 Windows Form 程序创建技巧	10
1.2.9 第一个 ASP.NET 程序	11
1.2.10 创建 ASP.NET 程序	11
1.2.11 Web 程序和 Windows 程序的区别	14
1.3 一个优秀的开发助手——MSDN	15
1.4 小结	15

第二篇 C#语法及面向对象基础

第 2 章 C#语法基础

18

每种计算机语言都有其基本的语法规则,用来管理内存和处理数据。任何计算机程序都必须遵循其编写语言的语法规则。C#语言继承了许多 C++的语法特性,简化了 C++语法的诸多复杂

性。很多关键字和 C++中的是一样的。这对使用过 C、C++、Java 等语言的读者来说是个好消息，但是 C#语法并不是 C++语法的简单复制，如它对数据类型的管理更为严格。

2.1	C#的基本语法	18
2.2	变量	19
2.2.1	变量的声明	19
2.2.2	变量的赋值	20
2.2.3	简单数据类型	21
2.2.4	使用简单数据类型	22
2.2.5	使用 struct 创建结构类型	24
2.2.6	结构类型例程	25
2.2.7	定义结构的构造函数	27
2.2.8	类型转换	29
2.2.9	隐式转换	29
2.2.10	显式转换	31
2.2.11	根据参与类型转换的划分	33
2.3	常量	33
2.3.1	静态常量	33
2.3.2	动态常量	34
2.3.3	使用动态常量	35
2.4	表达式	36
2.4.1	数学运算符	37
2.4.2	普通数学运算符	37
2.4.3	自加和自减运算符	38
2.4.4	赋值运算符	39
2.4.5	比较运算符	41
2.4.6	运算符的优先级	42
2.4.7	命名空间	43
2.4.8	嵌套命名空间	44
2.5	流程控制	45
2.5.1	分支语句	45
2.5.2	三元运算符	45
2.5.3	if 语句	47
2.5.4	使用 if 语句	48
2.5.5	程序流程	49
2.5.6	switch 语句	50
2.5.7	使用 switch 语句	50

2.5.8	goto 语句	51
2.5.9	循环语句	53
2.5.10	do-while 语句	53
2.5.11	使用 do-while 语句	53
2.5.12	while 语句	55
2.5.13	使用 while 语句	55
2.5.14	for 语句	57
2.5.15	使用 for 语句	58
2.5.16	foreach 循环语句	59
2.5.17	循环中断语句	61
2.5.18	使用 break 语句	61
2.5.19	使用 continue 语句	62
2.5.20	使用 return 语句	63
2.6	小结	64

第 3 章 C#面向对象基础

65

面向对象编程的英文简称是 OOP (Object Oriented Programming) , 该项技术是目前运用最广泛的程序化设计方法, 几乎已经完全取代了过去的面向过程编程。C#从一诞生开始, 就是为面向对象编程所准备的。类是面向对象编程的核心部件, 它描述了一组具有相同特性和行为的对象。基于面向对象的应用程序, 就是由几个或几十个甚至更多的类组成, 且类之间总是保持着或多或少的关系。类其实也是数据类型, 所以在面向对象编程中程序员可以自定义数据类型, 这和面向过程编程是有本质区别的。

3.1	类的基本概念	65
3.1.1	C#中的类定义	65
3.1.2	字段	66
3.1.3	常量	66
3.1.4	域	66
3.1.5	类的方法	67
3.1.6	类的属性	69
3.1.7	类的索引器	71
3.1.8	类的构造函数和析构函数	72
3.1.9	事件	74
3.2	Visual Studio 中的类向导	74
3.3	事件和委托	77
3.3.1	委托	78
3.3.2	委托的事件处理程序	79

3.3.3 委托中的 GUI 事件	82
3.4 面向对象的特征	83
3.4.1 继承	83
3.4.2 多态	85
3.4.3 封装	87
3.4.4 重载	87
3.4.5 方法重载	88
3.4.6 运算符重载	88
3.5 接口	89
3.5.1 接口的声明	89
3.5.2 接口的使用	90
3.6 面向对象的其他主题	90
3.6.1 命名空间	91
3.6.2 程序集	92
3.6.3 类库	93
3.7 小结	94

第 4 章 字符串处理

95

字符串是 C# 中应用较广的一个概念，它作为 C# 中最为常见的数据处理类型之一，能和其他数据类型进行相互转化。NET Framework 中有两个用于字符串处理的基本类——String 类和 StringBuilder 类。本章从这两个类开始，陆续介绍字符串的一些相关操作。

4.1 字符串的常用功能	95
4.1.1 字符串的比较	95
4.1.2 字符串的格式化	96
4.1.3 字符串的大小写转换	96
4.1.4 字符串的拆分与截取	96
4.2 StringBuilder 类	97
4.2.1 字符串对象的追加	97
4.2.2 字符串的插入、删除与替换	97
4.3 正则表达式	98
4.3.1 System.Text.RegularExpressions 命名空间	98
4.3.2 正则表达式的语法	98
4.3.3 正则表达式验证用户输入	99
4.3.4 使用正则表达式来替换字符串	100
4.4 小结	101

第5章 数组处理

102

数组同上一章讲的字符串一样，也是 C# 数据处理方面的一个基础内容。数组是一种基本的数据类型，和其他类型（比如 int 型、float 型、double 型、string 型等）的区别是它能存储一组数据，而非单个数据。这个性质使得数组的应用十分广泛。

5.1 创建数组	102
5.2 多维数组	103
5.3 动态数组	103
5.3.1 创建简单的动态数组	104
5.3.2 动态数组的排序	105
5.4 数组的特殊操作	106
5.4.1 作为参数传递的数组	106
5.4.2 将一维数组作为参数传递	106
5.4.3 将多维数组作为参数传递	107
5.4.4 数组的数组	108
5.5 小结	108

第6章 集合与泛型

109

集合与数组的基本功能大致相同，但它在处理数据时具有更强更灵活的功能。.NET Framework 提供了用于数据存储和检索的专用集合类，包含在 System.Collections 和 System.Collections.Generic（泛型）中。泛型是 C# 2008 的一个新增特性，通过泛型可以定义更安全的数据结构，得到更优质的代码。在本章中，将对集合与泛型进行详细介绍。

6.1 什么是集合	109
6.2 SortedList 可排序数组集合	109
6.3 Queue 消息队列集合	110
6.4 Stack 栈集合	112
6.5 Hashtable 哈希表集合	113
6.6 集合中的一些重要概念	114
6.6.1 集合中的索引器	114
6.6.2 集合中的迭代器	115
6.6.3 深度复制	116
6.7 为集合使用泛型	118
6.7.1 定义泛型类	118
6.7.2 定义泛型接口	120
6.7.3 定义泛型方法	120
6.7.4 定义泛型委托	121
6.8 小结	121

代码调试是编程中必不可少的一个环节,对于任何一种语言都不例外。无论多么优秀的程序员,也不能保证写出的代码永远都不出错。一般来讲,程序的错误分为语法错误和逻辑错误。语法错误可能是大小写不对,对象名拼写错误等,这些错误一般是发生在程序编译期间,较容易被检查出来;而逻辑错误一般发生在程序运行期间,有时候是很难被发现的。此时,Visual Studio 2008 强大的代码调试功能就能发挥作用了。C#中的代码调试分为两个方面,一方面是 Visual Studio 2008 本身的调试,它提供了一些窗口,能实时监控程序的运行与走向;另一方面是运用附加代码对程序进行监控,如 try...catch 模块。本章将对以上内容进行详细介绍。

7.1 Visual Studio 2008 中的调试	122
7.1.1 正常模式下的调试	122
7.1.2 中断模式下的调试	124
7.2 错误和异常处理	127
7.2.1 使用 try...catch...finally 语句	127
7.2.2 常见异常类型	129
7.2.3 创建和抛出自定义的异常	129
7.3 小结	130

第三篇 窗体应用程序开发

前面各章内容基本上都使用了控制台应用程序作为演示例程。从本章开始,将介绍大量有关 Windows 窗体应用程序的开发知识和技巧。这也是一件令人兴奋的事情,因为 Windows 窗体应用程序有着直观的界面和非常强大的功能。基本上可以实现开发人员所希望的任何功能。借助窗体设计器等工具,VS2008 为快速开发友好、交互及个性化的 Windows 窗体应用程序,提供了强大支持。

8.1 一个 Windows 窗体程序的深入分析	132
8.1.1 一个典型的 Windows 窗体程序	132
8.1.2 调试 Windows 窗体程序	134
8.1.3 调试总结	137
8.2 Windows 窗体基本设计内容	137
8.2.1 设置窗体的位置与大小	137
8.2.2 人工设置窗体的位置与大小	138
8.2.3 设置窗体的位置与大小的其他方法	139
8.2.4 设置窗体的边框	140
8.2.5 使用属性窗口设置窗体边框	140
8.2.6 在代码中设置各种窗体边框	141

8.2.7	设置控件框和最小化、最大化、关闭按钮	142
8.2.8	在程序中设置最小化、最大化、关闭按钮	142
8.2.9	设置控件框按钮	144
8.2.10	窗体的事件	145
8.2.11	Load 事件	145
8.2.12	鼠标悬停事件	146
8.2.13	按键事件	147
8.2.14	鼠标单击事件	147
8.3	SDI 应用程序、MDI 应用程序和菜单	148
8.3.1	基于对话框的应用程序	148
8.3.2	单文档 (SDI) 应用程序	148
8.3.3	多文档 (MDI) 应用程序	149
8.3.4	菜单	149
8.3.5	多文档、单文档和菜单的设计方法	150
8.4	快捷菜单 (ContextMenuStrip)	154
8.5	小结	156

第 9 章 使用 Windows 窗体控件

157

Windows 窗体控件是包含在窗体对象中的对象。VS2008 提供了许多窗体控件，大部分窗体控件具有用户界面的特性，即在程序运行时可以看到。当然，也有一部分控件是没有用户界面特性的，称为不可视控件，如前面章节介绍的 MenuStrip 菜单控件、ContextMenuStrip 控件等。

9.1	控件	157
9.1.1	控件的属性	157
9.1.2	使用 Dock 属性布局控件	158
9.1.3	控件的事件	159
9.1.4	为控件添加事件	160
9.2	TextBox、RichTextBox 和 Button 控件	161
9.2.1	使用 Button 控件实现按钮功能	161
9.2.2	为 Button 控件设置图片和显示样式	162
9.2.3	使用 TextBox 控件实现文本的输入和输出功能	163
9.2.4	一个 Button 和 TextBox 控件的综合实例	165
9.2.5	综合实例说明	170
9.3	Label 和 LabelLink 控件	171
9.3.1	使用 Label 控件实现标签功能	171
9.3.2	使用 LinkLabel 控件实现超链接设置	172
9.4	RadioButton 和 CheckBox 控件	174

9.4.1	使用 RadioButton 控件实现单选功能	174
9.4.2	使用 CheckBox 控件实现复选功能	174
9.4.3	一个相应单选和复选事件的实例	175
9.5	使用 GroupBox 和 Panel 控件实现控件的分组	178
9.6	ListView 控件	179
9.6.1	设置 ListView 控件的属性	179
9.6.2	ListView 控件的事件	180
9.6.3	一个读写 ListView 控件的实例	181
9.6.4	添加窗体的加载事件处理程序	183
9.6.5	程序运行效果	185
9.6.6	读取 XML 数据并填充 ListView	186
9.6.7	VS2008 中建立 XML	186
9.6.8	VS2008 中读取 XML 并填充 ListView	187
9.6.9	将 ListView 数据写入 XML	190
9.6.10	AfterLabelEdit 事件	190
9.6.11	ListView 各项的修改	193
9.7	TreeView 控件	197
9.7.1	TreeView 的属性和事件	197
9.7.2	TreeView 树读取并显示 XML 节点	199
9.7.3	程序运行结果及说明	202
9.8	小结	202

第 10 章 GDI+概述

203

GDI+是 GDI 的下一代版本, 全称是 Graphics Device Interface+ (图形设备接口), 主要用于 .NET Framework 中的绘图。图形总能给人一种直观和清晰的印象, 在应用程序中, 运用图形进行表达往往能起到很好的效果。图形代码的编写是开发者比较喜爱的一种方式, 它简单且能立刻看到效果。GDI+是一个类库, 里面封装了一些关于绘图的命名空间和类, 它让开发者能创建图形, 绘制文本以及将图形或图像作为对象来处理。

10.1	GDI+的工作原理	203
10.2	图形绘制基础	204
10.2.1	Graphics 类	204
10.2.2	Pen 类的用法	205
10.2.3	Brush 类的用法	207
10.2.4	Font 类的用法	210
10.3	GDI+的一些实例	211
10.3.1	图像的缩放	212

10.3.2 图像的剪切	214
10.3.3 图像的旋转	216
10.3.4 图像的切换显示	218
10.3.5 图片的打印和打印预览	219
10.3.6 自己写播放器	221
10.4 小结	223

第 11 章 文件的操作

225

文件处理是编程语言中最重要的功能之一，也是 C# 语言中最重要的功能之一，这使得 C# 能够支持常用于处理大量持久数据的商用程序。本章将讨论 C# 强大且丰富的文件处理和流输入/输出功能。

11.1 流的概念	225
11.2 用于输入和输出的类	226
11.2.1 文件的移动、删除和复制	226
11.2.2 目录的移动、删除和复制	228
11.2.3 流和文件的读写	228
11.3 对话框	233
11.3.1 文件对话框	233
11.3.2 字体对话框	237
11.3.3 颜色对话框	239
11.3.4 页面设置	241
11.3.5 打印预览对话框和打印对话框	242
11.4 高级主题	245
11.4.1 监控文件结构	245
11.4.2 文件异步写入	250
11.5 小结	252

第 12 章 部署 Windows 应用程序

253

部署的含义是指将已完成的应用程序或组件，安装到其他计算机上的过程。在 .NET Framework 中，有两个部署选项供用户选择，包括 Windows Installer 和 ClickOnce。其中，Windows Installer 是使用较早的一种部署方式，它允许用户创建安装包并分发给其他用户，拥有此安装包的用户，只要按提示进行操作即可完成程序的安装，Windows Installer 在中小程序的部署中应用十分广泛。ClickOnce 是 VS2008 中的重要部署方式，它允许用户将 Windows 应用程序，发布到 Web 服务器或网络共享文件夹，允许其他用户进行在线安装。本章将主要介绍 Windows 应用程序的部署，有关 Web 应用程序部署的内容将在本书后面部分讲解。

12.1 Windows Installer 部署	253
---------------------------------	-----

12.1.1	创建部署	253
12.1.2	在部署中编辑器	255
12.1.3	文件系统编辑器	255
12.1.4	用户界面编辑器	257
12.1.5	文件类型编辑器	260
12.1.6	启动条件编辑器	261
12.1.7	注册表编辑器	262
12.1.8	自定义操作编辑器	262
12.2	Windows Installer 部署安装全过程	263
12.2.1	欢迎	263
12.2.2	自述文件	264
12.2.3	许可协议	264
12.2.4	客户信息	264
12.2.5	注册用户和复选框	265
12.2.6	选择安装文件夹和确认安装	265
12.2.7	进度和安装提示	265
12.2.8	安装完成	266
12.3	程序的卸载和修复	267
12.4	VS2005 中的其他打包方式	268
12.5	ClickOnce 部署	268
12.5.1	部署策略的选择	269
12.5.2	创建 ClickOnce 部署	269
12.5.3	安装全过程	271
12.6	两种部署方式的比较	272
12.7	小结	272

第四篇 C#数据存取

第 13 章 数据库基础知识

274

数据库的发展历史很悠久，从 20 世纪 60 年代开始出现的网状、层次数据库系统，到今天的以面向对象为特征的数据库系统，数据库已经经历了数次变革。数据库是利用计算机技术统一管理的相关数据的集合，它能动态地存储大量的相互关联的数据。Microsoft SQL Server 2005 是本章将要重点介绍的内容，它是微软推出的最新数据库，它较之前的 SQL 数据库系列在很多方面都有改进，如全文检索、查询引擎、统计信息等。因为本书的重点不在数据库本身，所以对这些细节就不再详述。

13.1	SQL Server 2005 的安装	274
------	---------------------	-----

13.2	Transact-SQL 语言	277
13.2.1	Transact-SQL 的特点	277
13.2.2	数据控制语言	277
13.2.3	数据定义语言	278
13.2.4	数据操纵语言	278
13.2.5	事务管理语言	278
13.2.6	其他附加语言	278
13.3	表	283
13.3.1	创建、修改和删除表	283
13.3.2	操纵表中的数据	284
13.3.3	主键约束	286
13.3.4	外键约束	287
13.3.5	唯一性约束	288
13.3.6	CHECK 约束	288
13.3.7	默认约束	288
13.4	存储过程	288
13.4.1	创建存储过程	289
13.4.2	执行和删除存储过程	290
13.4.3	SQL Server 2005 中的存储过程向导	290
13.5	触发器	292
13.5.1	创建和使用触发器	292
13.5.2	SQL Server 2005 中的触发器向导	293
13.6	事务和锁	294
13.6.1	事务	294
13.6.2	锁	296
13.7	小结	297

第 14 章 利用 ADO.NET 访问数据库

298

ADO.NET 是活动数据对象 (Active Data Objects, ADO) 的更新版本, 它主要用于访问数据。它支持三种数据访问格式, 包括连接的数据访问、断开的数据访问和 XML 访问。ADO.NET 解决了应用程序和数据库之间不能分离的问题, 它让程序员能轻松地连接到数据库, 并在数据库中进行检索、添加、删除、修改和插入等操作。它主要由两个部分构成, 即 DataSet 和 .NET 数据提供者, 其中 DataSet 用于以表格的形式存储数据; 而 .NET 数据提供者能允许程序员与单独的数据源通信, 常用的数据提供者包括 ODBC、SQL Server 和 OLE DB。

14.1	ADO.NET 中的图形化方式	298
------	-----------------	-----

14.1.1	连接到数据库	298
--------	--------	-----