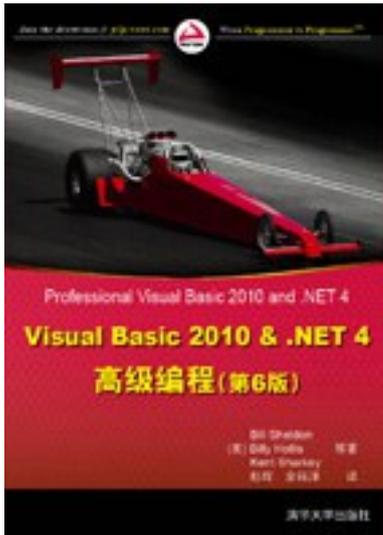


Visual Basic 2010&.NET 4高级编程



[Visual Basic 2010&.NET 4高级编程_下载链接1](#)

著者:[美]Bill Sheldon

出版者:清华大学出版社

出版时间:2011-6

装帧:

isbn:9787302254508

visual basic

2010引入激动人心的新特性和功能，巩固了这个真正的面向对象语言的地位，并提供了对更好的新技术的访问。《visual basic 2010 &.net 4 高级编程(第6版)》物超所值，完整地介绍了visual basic的内容，讨论了visual basic 2010新改进。本书的专家作者团队解释了visual basic 2010如何与net4合并起来，使用wpf、windows窗体、vsto等构建应用程序，以及如何建立基于wcf、wf和sharepoint的应用程序和库。

主要内容

- 介绍核心语言元素和开发工具
- 研究面向对象编程和并行编程

- 讨论异常处理、调式和跟踪日志，以及visual basic 2010中的历史调试功能
- 描述用于包含、访问和传送数据的常用结构和工具
- 关注如何为客户端和基于web的解决方案创建组件化的应用程序
- 理解最新的应用程序开发技术:silverlight、sharepoint、wcf、workflow和wpcf

作者介绍:

Bill Sheidon，是一位Visual Basic的Microsoft MVP和InterKnowlogy的首席工程师。 Bill Hollis，是Microsoft开发关系的一位MSD区域主管。 Kent Sharkey，是一位独立的软件顾问，以及MSDN的前内容战略家。 Jonathan Marbutt是WayCool软件开发公司的副总裁。 Rob Windsor是ObjectSharp咨询公司的资深顾问，是一位Microsoft MVP。 Gaston C.Hillar是一位独立的软件顾问，擅长并行化和多核心技术。

目录: 第 I 部分 语言结构和环境

第1章 Visual Studio 2010 3

1.1 Visual Studio 2010: 从Express到Ultimate的各种版本 4

1.2 Visual Basic的关键字和语法 7

1.2.1 控制台应用程序 10

1.2.2 从项目模板上创建项目 11

1.2.3 Solution Explorer窗口 13

1.2.4 项目属性 14

1.2.5 Assembly Information屏幕 15

1.2.6 编译设置 16

1.2.7 调试属性 20

1.2.8 引用 21

1.2.9 资源 23

1.2.10 设置 24

1.2.11 其他项目属性选项卡 26

1.3 ProVB_VS2010项目 27

1.3.1 在代码中设置窗体属性 29

1.3.2 Visual Studio的其他组件 37

1.4 增强示例应用程序的功能 37

1.4.1 定制代码 39

1.4.2 构建应用程序 43

1.4.3 重用第一个Windows窗体 50

1.5 Visual Studio 2010中的特色

功能 51

1.5.1 构建配置 51

1.5.2 任务列表 53

1.5.3 Command窗口 54

1.5.4 Server Explorer 54

1.5.5 在Visual Studio 2010中

记录和使用宏 55

1.5.6 类图 57

1.5.7 应用程序生命周期管理 58

1.5.8 性能工具 60

1.6 小结 62

- 第2章 对象和Visual Basic 63
 - 2.1 面向对象的术语 64
 - 2.1.1 对象、类和实例 64
 - 2.1.2 对象的组成 65
 - 2.1.3 System.Object 68
 - 2.2 使用Visual Basic类型 68
 - 2.2.1 值类型和引用类型 69
 - 2.2.2 基本类型 71
 - 2.3 命令：条件语句 72
 - 2.3.1 If Then 73
 - 2.3.2 比较运算符 73
 - 2.3.3 Select Case 75
 - 2.4 值类型(结构) 75
 - 2.4.1 布尔类型 76
 - 2.4.2 整数类型 77
 - 2.4.3 无符号类型 78
 - 2.4.4 小数类型 78
 - 2.4.5 Char和Byte类型 81
 - 2.4.6 DateTime类型 81
 - 2.5 引用类型(类) 82
 - 2.5.1 Object类 82
 - 2.5.2 String类 83
 - 2.5.3 XML字面量 87
 - 2.5.4 DBNull类和IsDBNull()
函数 89
 - 2.6 参数传递 89
 - 2.7 变量的作用域 91
 - 2.8 使用对象 92
 - 2.8.1 对象的声明和实例化 92
 - 2.8.2 对象引用 93
 - 2.8.3 取消对象的引用 93
 - 2.8.4 前期绑定与后期绑定 94
 - 2.9 数据类型转换 95
 - 2.10 创建类 100
 - 2.10.1 类 100
 - 2.10.2 事件的处理 110
 - 2.10.3 处理多个事件 111
 - 2.10.4 WithEvents关键字 111
 - 2.10.5 触发事件 111
 - 2.10.6 声明和触发定制事件 112
 - 2.10.7 用WithEvents关键字
接收事件 113
 - 2.10.8 用AddHandler
接收事件 115
 - 2.10.9 构造函数方法 116
 - 2.10.10 终止和清除 117
 - 2.11 高级概念 118
 - 2.11.1 重载方法 119
 - 2.11.2 重载构造函数方法 121
 - 2.11.3 共享方法、变量和
事件 122
 - 2.11.4 运算符重载 127
 - 2.11.5 委托 129
 - 2.11.6 类和组件 133

- 2.11.7 Lambda表达式 134
- 2.12 小结 135
- 第3章 定制对象 137
 - 3.1 继承 138
 - 3.1.1 继承的实现 139
 - 3.1.2 继承的层次 157
 - 3.1.3 与基类、类及对象交互 159
 - 3.1.4 构造函数 164
 - 3.1.5 Protected作用域 169
 - 3.1.6 事件与继承 171
 - 3.1.7 共享方法 174
 - 3.1.8 共享事件 176
 - 3.1.9 创建抽象基类 176
 - 3.2 多接口 178
 - 3.2.1 对象接口 178
 - 3.2.2 辅助接口 180
 - 3.3 抽象性 185
 - 3.4 封装性 188
 - 3.5 多态性 190
 - 3.5.1 方法签名 190
 - 3.5.2 实现多态性 191
 - 3.6 进一步讨论继承 200
 - 3.7 小结 211
- 第4章 公共语言运行库 213
 - 4.1 .NET应用程序的组成元素 214
 - 4.1.1 模块 214
 - 4.1.2 程序集 215
 - 4.1.3 类型 215
 - 4.2 版本化与部署 216
 - 4.2.1 对版本化更好的支持 216
 - 4.2.2 Major.Minor.Build.Revision
版本介绍 217
 - 4.2.3 更好的部署 217
 - 4.3 跨语言集成 218
 - 4.3.1 通用类型系统 218
 - 4.3.2 元数据 219
 - 4.3.3 对元数据更好的支持 219
 - 4.3.4 属性 220
 - 4.3.5 Reflection API 222
 - 4.4 IL反汇编程序 222
 - 4.5 内存管理 223
 - 4.5.1 传统的垃圾回收机制 223
 - 4.5.2 更快地为对象分配内存 230
 - 4.5.3 垃圾回收器的优化 231
 - 4.6 名称空间 232
 - 4.6.1 名称空间的概念 233
 - 4.6.2 名称空间与引用 236
 - 4.6.3 常用的名称空间 237
 - 4.6.4 导入名称空间并指定别名 239
 - 4.6.5 为名称空间指定别名 240
 - 4.6.6 在ASP.NET中引用名称
空间 241
 - 4.7 创建自己的名称空间 241
 - 4.8 My关键字 244

- 4.8.1 My.Application名称空间 244
- 4.8.2 My.Computer名称空间 248
- 4.8.3 My.Forms名称空间 251
- 4.8.4 My.Resources名称空间 251
- 4.8.5 My.User名称空间 251
- 4.8.6 My.WebServices名称空间 251
- 4.9 扩展My名称空间 252
- 4.10 小结 254
- 第5章 用Visual Basic进行声明式编程 255
 - 5.1 声明式编程与Visual Basic 256
 - 5.2 使用XAML创建窗口 257
 - 5.3 XAML语法 260
 - 5.3.1 XAML语言基础 261
 - 5.3.2 使用XAML声明工作流 264
 - 5.4 小结 265
- 第6章 异常处理和调试 267
 - 6.1 Visual Studio 2010 Team System
的新增内容：历史调试 267
 - 6.2 与Visual Basic 6兼容的
注意事项 268
 - 6.3 .NET中的异常处理 268
 - 6.4 结构化异常处理的关键字 269
 - 6.4.1 Try、Catch和Finally
关键字 270
 - 6.4.2 Throw关键字 271
 - 6.4.3 抛出新的异常 272
 - 6.4.4 Exit Try语句 273
 - 6.4.5 嵌套的Try结构 274
 - 6.4.6 异常属性的使用 275
 - 6.4.7 Message属性 276
 - 6.4.8 InnerException和TargetSite
属性 276
 - 6.5 与Visual Basic 6样式的
错误处理交互操作 280
 - 6.6 记录错误 281
 - 6.6.1 事件日志 281
 - 6.6.2 事件、方法和属性 282
 - 6.6.3 写入跟踪文件 284
 - 6.7 小结 286
- 第7章 测试驱动的开发 287
 - 7.1 测试的内容和方式 288
 - 7.2 Visual Studio中的TDD
工具 290
 - 7.3 单元测试过程 291
 - 7.3.1 创建测试程序 291
 - 7.3.2 运行测试程序 294
 - 7.3.3 测试数据访问代码 295
 - 7.3.4 使用Generate from Usage
特性 302
 - 7.4 其他Visual Studio版本 306
 - 7.5 第三方测试框架 306
 - 7.6 小结 307
- 第II部分 业务对象和数据访问

第8章 数组、集合和泛型	311
8.1 数组	312
8.1.1 多维数组	313
8.1.2 UBound函数	314
8.1.3 ReDim语句	314
8.1.4 Preserve关键字	315
8.2 集合	315
8.2.1 循环语句	317
8.2.2 装箱	319
8.3 泛型	320
8.3.1 泛型的使用	321
8.3.2 Nullable类型	322
8.3.3 泛型类型	323
8.3.4 泛型方法	326
8.4 创建泛型	327
8.4.1 泛型类型	328
8.4.2 泛型方法	334
8.4.3 约束	335
8.4.4 泛型和后期绑定	338
8.4.5 协变和逆变	339
8.5 小结	340
第9章 在VB中使用XML	341
9.1 XML简介	342
9.2 XML序列化	343
9.3 System.Xml文档支持	348
9.4 XML流样式分析程序	348
9.4.1 写入XML流	349
9.4.2 读取XML流	352
9.4.3 文档对象模型(DOM)	360
9.5 XSLT转换	364
9.5.1 使用XSLT转换不同的XML标准	367
9.5.2 System.Xml.Xsl中定义的其他类和接口	370
9.6 ASP.NET中的XML	370
9.6.1 XmlDataSource服务器控件	370
9.6.2 XmlDataSource控件的名称空间问题	374
9.6.3 Xml服务器控件	375
9.7 LINQ to XML	376
9.8 LINQ to XML帮助对象	376
9.8.1 XDocument对象	377
9.8.2 XElement对象	377
9.8.3 XNamespace对象	378
9.8.4 XAttribute对象	380
9.9 Visual Basic和XML字面量	381
9.10 使用LINQ查询XML文档	382
9.10.1 查询静态的XML文档	382
9.10.2 查询动态的XML文档	384
9.11 处理XML文档	385
9.11.1 读取XML文档	385
9.11.2 写入XML文档	386
9.12 VB中的Lambda表达式	387

- 9.13 小结 389
- 第10章 ADO.NET和LINQ 391
 - 10.1 ADO.NET的体系结构 392
 - 10.2 ADO.NET的基本功能 393
 - 10.2.1 ADO.NET的常见任务 393
 - 10.2.2 ADO.NET的基本名称空间和类 398
 - 10.2.3 ADO.NET组件 399
 - 10.3 .NET数据提供程序 400
 - 10.3.1 Connection对象 400
 - 10.3.2 Command对象 401
 - 10.3.3 通过Command对象使用存储过程 402
 - 10.3.4 DataReader对象 405
 - 10.3.5 命令的异步执行 407
 - 10.3.6 DataAdapter对象 409
 - 10.3.7 SQL Server .NET数据提供程序 413
 - 10.3.8 OLE DB .NET数据提供程序 413
 - 10.4 DataSet组件 413
 - 10.4.1 DataTableCollection对象 414
 - 10.4.2 DataRelationCollection对象 414
 - 10.4.3 ExtendedProperties属性 414
 - 10.4.4 创建和使用DataSet对象 415
 - 10.4.5 ADO.NET的DataTable对象 417
 - 10.4.6 DataSet和DataTable对象的高级ADO.NET特性 418
 - 10.5 使用通用提供程序模型 420
 - 10.6 ADO.NET中的连接池 422
 - 10.7 Transactions类和System.Transactions名称空间 423
 - 10.7.1 创建事务 423
 - 10.7.2 创建资源管理器 425
 - 10.8 LINQ to SQL 425
 - 10.9 LINQ to SQL和Visual Basic 426
 - 10.9.1 用LINQ to SQL提取数据：创建控制台应用程序 426
 - 10.9.2 O/R设计器 427
 - 10.9.3 创建Product对象 428
 - 10.10 对象到LINQ对象的映射 429
 - 10.10.1 DataContext对象 430
 - 10.10.2 Table(TEntity)对象 432
 - 10.11 查询数据库 433
 - 10.11.1 使用查询表达式 433

- 10.11.2 查询表达式详述 433
- 10.11.3 用表达式过滤 434
- 10.11.4 联接 434
- 10.11.5 数据项的组合 435
- 10.12 存储过程 437
- 10.13 更新数据库 438
- 10.14 小结 440
- 第11章 使用Entity Framework访问数据 441
 - 11.1 对象关系映射 441
 - 11.2 Entity Framework体系结构 442
 - 11.2.1 概念模型 443
 - 11.2.2 存储模型 446
 - 11.2.3 映射模型 447
 - 11.2.4 LINQ to Entities 448
 - 11.2.5ObjectContext 449
 - 11.3 把对象映射到实体上 451
 - 11.3.1 简单映射 451
 - 11.3.2 对多个对象使用一个表 453
 - 11.3.3 对一个对象使用多个表 455
 - 11.4 从模型中生成数据库 457
 - 11.5 小结 460
- 第12章 使用SQL Server 461
 - 12.1 SQL Server Compact 462
 - 12.1.1 连接SQL Server Compact Edition数据库 463
 - 12.1.2 同步数据 466
 - 12.2 SQL Server内置的XML功能 472
 - 12.3 SQL Server中的CLR集成 474
 - 12.3.1 决定使用T-SQL还是VB 475
 - 12.3.2 创建用户定义的类型 475
 - 12.3.3 创建存储过程 487
 - 12.3.4 在SQL Server中使用Web服务 493
 - 12.3.5 SQL Server 2008特性 498
 - 12.4 WCF数据服务 499
 - 12.4.1 REST 499
 - 12.4.2 Atom和JSON 499
 - 12.4.3 使用WCF数据服务提供数据 500
 - 12.4.4 WCF数据服务的客户端库 504
 - 12.5 小结 508
- 第13章 服务(XML/WCF) 509
 - 13.1 服务 510
 - 13.1.1 网络角度 510
 - 13.1.2 应用程序的发展 510
 - 13.1.3 合并网络和应用程序

- 开发 510
- 13.1.4 Web服务基础 511
- 13.1.5 存在的问题 512
- 13.1.6 其他技术 512
- 13.1.7 Web服务 513
- 13.1.8 组合起来 514
- 13.1.9 WCF服务的构成 514
- 13.2 向SOA迈出一大步 515
- 13.2.1 WCF的功能 516
- 13.2.2 协定和元数据 516
- 13.2.3 使用WS-*协议 517
- 13.3 建立WCF服务 518
- 13.4 建立WCF使用者应用程序 524
- 13.4.1 添加服务引用 525
- 13.4.2 查看引用 526
- 13.4.3 配置文件的修改 529
- 13.4.4 编写使用者应用程序的代码 531
- 13.5 使用数据协定 533
- 13.6 名称空间 535
- 13.6.1 建立主机应用程序 535
- 13.6.2 建立使用者应用程序 536
- 13.6.3 查看HelloCustomer Service的WSDL和架构 538
- 13.7 小结 540
- 第Ⅲ部分 智能客户端应用程序
- 第14章 Windows窗体 543
- 14.1 System.Windows.Forms名称空间 543
- 14.2 窗体的使用 544
- 14.2.1 设置启动窗体 544
- 14.2.2 通过Sub Main显示窗体 545
- 14.2.3 Application类的更多内容 545
- 14.2.4 窗体的启动位置 545
- 14.2.5 窗体边框 545
- 14.2.6 始终置顶——TopMost属性 546
- 14.2.7 附属窗体 546
- 14.2.8 改变窗体的透明度 547
- 14.2.9 可视化继承 549
- 14.2.10 滚动窗体 549
- 14.2.11 MDI窗体 549
- 14.2.12 VB 2010中的MDI 549
- 14.2.13 对话框窗体 551
- 14.2.14 运行时的窗体 553
- 14.2.15 默认的窗体实例 554
- 14.3 控件 554
- 14.3.1 以Tab键切换控件的顺序 554

- 14.3.2 所有控件的属性 555
- 14.3.3 动态调整控件的大小和布局 555
- 14.3.4 FlowLayoutPanel 控件 557
- 14.3.5 TableLayoutPanel 控件 558
- 14.3.6 Panel和GroupBox容器 控件 559
- 14.3.7 扩展的Provider控件 560
- 14.3.8 数据输入的高级功能 562
- 14.3.9 验证数据输入 564
- 14.3.10 工具栏与ToolStrip 控件 565
- 14.3.11 菜单 568
- 14.3.12 通用对话框 569
- 14.3.13 拖放操作 571
- 14.3.14 标准Windows窗体 控件小结 573
- 14.3.15 处理相关控件组 575
- 14.3.16 在运行时添加控件 576
- 14.4 其他编程技巧 577
- 14.5 小结 577
- 第15章 Windows窗体的高级功能 579
- 15.1 在可视化控件中封装逻辑 579
- 15.2 在Windows窗体中开发自定义的控件 580
 - 15.2.1 继承现有的控件 580
 - 15.2.2 构建复合控件 580
 - 15.2.3 从头编写控件 581
- 15.3 继承现有的控件 581
 - 15.3.1 基本步骤 581
 - 15.3.2 给派生的控件添加代码 581
 - 15.3.3 其他有用的特性 584
 - 15.3.4 为派生的控件自定义事件 585
 - 15.3.5 限制选中项数的CheckedListBox 586
- 15.4 Control与UserControl 基类 589
 - 15.4.1 Control类 589
 - 15.4.2 UserControl类 589
- 15.5 复合控件 590
 - 15.5.1 创建复合的UserControl 591
 - 15.5.2 改变控件的大小 591
 - 15.5.3 提供子控件的属性 592
 - 15.5.4 一个具体的例子 592
- 15.6 从头构建控件 595
- 15.7 给工具箱中的控件添加图标 600
- 15.8 在控件中嵌入其他控件 601

- 15.9 小结 602
- 第16章 集成WPF和Windows窗体的用户控件 605
 - 16.1 集成库 606
 - 16.2 在Windows窗体中包含WPF控件 607
 - 16.2.1 创建WPF控件库 608
 - 16.2.2 Windows窗体应用程序 610
 - 16.3 在WPF中包含Windows窗体控件 616
 - 16.4 集成的限制 621
 - 16.5 小结 622
- 第17章 WPF桌面应用程序 623
 - 17.1 内容、位置、原因、方式——WPF策略 624
 - 17.2 光栅图形和矢量图形 625
 - 17.3 下一个Windows项目应使用WPF吗 625
 - 17.4 创建WPF应用程序 626
 - 17.4.1 实现定制的WPF应用程序 627
 - 17.4.2 定制用户界面 639
 - 17.4.3 定制按钮 647
 - 17.4.4 WPF用户控件 651
 - 17.5 小结 672
- 第18章 Expression Blend 3 675
 - 18.1 了解Blend 676
 - 18.2 SketchFlow 682
 - 18.2.1 第一个SketchFlow 682
 - 18.2.2 SketchFlow Player 685
 - 18.2.3 归档SketchFlow 686
 - 18.3 小结 686
- 第19章 Silverlight 687
 - 19.1 什么是Silverlight 687
 - 19.1.1 Smooth Streaming 特性 688
 - 19.1.2 业界标准视频 688
 - 19.1.3 数字版权管理 688
 - 19.2 启动Silverlight项目 688
 - 19.2.1 Silverlight应用程序 689
 - 19.2.2 Silverlight导航应用程序 689
 - 19.2.3 Silverlight类库 690
 - 19.3 Silverlight解决方案 691
 - 19.3.1 Web应用程序 691
 - 19.3.2 应用程序库缓存 691
 - 19.3.3 Silverlight应用程序 692
 - 19.4 控件 695
 - 19.5 给Silverlight项目添加项 702
 - 19.5.1 Silverlight用户控件 703
 - 19.5.2 Silverlight应用程序类 703
 - 19.5.3 Silverlight页面 703
 - 19.5.4 Silverlight子窗口 703

- 19.5.5 Silverlight模板控件 703
- 19.5.6 Silverlight资源字典 704
- 19.6 浏览器之外的Silverlight 704
- 19.7 小结 705
- 第IV部分 Internet应用程序技术
- 第20章 Silverlight和服务 709
 - 20.1 服务和Silverlight 709
 - 20.1.1 ASMX Web服务 709
 - 20.1.2 WCF服务 712
 - 20.1.3 ADO.NET数据服务 716
 - 20.2 Model-View-ViewModel 725
 - 20.2.1 分割 725
 - 20.2.2 Model 725
 - 20.2.3 View 728
 - 20.2.4 ViewModel 729
 - 20.3 小结 729
- 第21章 使用ASP.NET 731
 - 21.1 ASP.NET的历史 731
 - 21.2 ASP.NET的重要特性 732
 - 21.2.1 开发效率 732
 - 21.2.2 性能和可伸缩性 732
 - 21.2.3 本地化 732
 - 21.2.4 健康监控 733
 - 21.2.5 易于访问数据 733
 - 21.2.6 管理和维护 733
 - 21.3 Visual Studio对ASP.NET的支持 733
 - 21.3.1 Web Site和Web Application项目 733
 - 21.3.2 ASP.NET应用程序文件夹 734
 - 21.3.3 Web服务器选项 735
 - 21.4 用Web窗体构建ASP.NET应用程序 735
 - 21.5 数据驱动的应用程序 746
 - 21.5.1 使用SqlDataSource控件绑定数据 746
 - 21.5.2 使用LinqDataSource控件绑定数据 754
 - 21.5.3 使用ObjectDastaSource控件绑定数据 757
 - 21.6 小结 759
- 第22章 ASP.NET的高级功能 761
 - 22.1 母版页 761
 - 22.1.1 创建母版页 762
 - 22.1.2 创建内容页 765
 - 22.1.3 为母版页提供默认内容 767
 - 22.2 导航 767
 - 22.2.1 使用SiteMapPath服务器控件 769
 - 22.2.2 Menu服务器控件 770
 - 22.3 使用ASP.NET的提供程序模型 771

- 22.4 成员和角色管理 776
- 22.5 配置文件属性 781
- 22.6 Microsoft Ajax(ASP.NET AJAX) 783
 - 22.6.1 理解对Ajax的需求 783
 - 22.6.2 Microsoft Ajax 的实现 784
 - 22.6.3 UpdatePanel控件和客户端服务调用 785
 - 22.6.4 示例项目 785
 - 22.6.5 添加UpdatePanel控件 789
 - 22.6.6 使用客户端服务调用和客户端模板 790
- 22.7 小结 795
- 第23章 ASP.NET MVC 797
 - 23.1 MVC和ASP.NET 798
 - 23.2 构建ASP.NET MVC应用程序 798
 - 23.2.1 创建项目 798
 - 23.2.2 控制器和操作 800
 - 23.2.3 添加模型 802
 - 23.2.4 视图 804
 - 23.2.5 路由 807
 - 23.2.6 搭框架和CRUD操作 808
 - 23.2.7 验证 815
 - 23.3 小结 817
- 第24章 SharePoint 2010开发 819
 - 24.1 简介 819
 - 24.1.1 SharePoint Foundation 2010 820
 - 24.1.2 SharePoint Server 2010 820
 - 24.1.3 SharePoint的术语 820
 - 24.1.4 SharePoint开发环境 821
 - 24.2 Feature和Solution Framework 821
 - 24.2.1 Feature 821
 - 24.2.2 Solution Framework 829
 - 24.3 用于SharePoint开发的Visual Studio工具 833
 - 24.4 SharePoint 2010对象模型 839
 - 24.4.1 服务器对象模型 840
 - 24.4.2 客户端对象模型 843
 - 24.5 构建Web 部件 845
 - 24.6 小结 851
- 第V部分 库和专业主题技术
- 第25章 Visual Studio Tools for Office 855
 - 25.1 VSTO的各个版本 856
 - 25.1.1 Office的自动化功能和VSTO 856
 - 25.1.2 免PIA部署 856
 - 25.1.3 VSTO项目类型 857
 - 25.2 Office业务应用程序的体系

- 结构 858
- 25.3 使用VBA和VSTO 859
- 25.4 创建文档模板(Word) 864
 - 25.4.1 给文档添加内容 866
 - 25.4.2 添加Ribbon和操作窗格 867
 - 25.4.3 激活操作窗格 870
 - 25.4.4 更新内容控件 872
- 25.5 创建Office插件(Excel) 875
- 25.6 Outlook Form Regions 881
- 25.7 小结 889
- 第26章 Windows Workflow Foundation 891
 - 26.1 应用程序中的工作流 891
 - 26.2 建立工作流 892
 - 26.2.1 用Windows Workflow Foundation添加工作流 892
 - 26.2.2 一个简单的工作流 894
 - 26.2.3 标准活动 897
 - 26.2.4 一个不太简单的工作流 899
 - 26.2.5 建立定制活动 907
 - 26.2.6 动态加载工作流 911
 - 26.3 重新构建工作流设计器 912
 - 26.4 小结 915
- 第27章 本地化 917
 - 27.1 文化和区域 917
 - 27.1.1 理解文化类型 918
 - 27.1.2 线程 919
 - 27.1.3 在ASP.NET中声明全局文化 921
 - 27.1.4 在ASP.NET中使用文化设置 922
 - 27.2 转换数值和操作 923
 - 27.2.1 理解日期之间的区别 923
 - 27.2.2 理解数字和货币的区别 925
 - 27.2.3 理解排序字符串的区别 927
 - 27.3 ASP.NET资源文件 929
 - 27.3.1 使用本地资源 929
 - 27.3.2 全局资源 933
 - 27.4 Windows窗体中的资源文件 935
 - 27.5 小结 938
- 第28章 与COM的交互操作 939
 - 28.1 理解COM 940
 - 28.2 COM和.NET的交互 940
 - 28.2.1 传统的组件 941
 - 28.2.2 .NET应用程序 942
 - 28.2.3 调试 945
 - 28.2.4 直接使用TlbImp 945
 - 28.2.5 后期绑定 946

- 28.3 ActiveX控件 950
 - 28.3.1 传统的ActiveX控件 950
 - 28.3.2 另一个.NET应用程序 952
 - 28.3.3 再次调试 954
- 28.4 在COM应用程序中使用.NET组件 954
 - 28.4.1 .NET组件 954
 - 28.4.2 RegAsm 956
 - 28.4.3 TlbExp 957
- 28.5 P/Invoke 957
- 28.6 小结 957
- 第29章 网络编程 959
 - 29.1 协议、地址和端口 959
 - 29.1.1 地址与计算机名 961
 - 29.1.2 端口：指定应用程序 961
 - 29.1.3 防火墙：不离不弃 962
 - 29.2 System.Net名称空间 963
 - 29.2.1 Web请求与响应 963
 - 29.2.2 使用WebClient简化常用的Web请求 969
 - 29.3 套接字 970
 - 29.3.1 构建应用程序 971
 - 29.3.2 创建Conversation窗口 973
 - 29.3.3 发送消息 980
 - 29.3.4 关闭应用程序 984
 - 29.4 在应用程序中使用Internet Explorer 988
 - 29.5 小结 991
- 第30章 应用程序服务 993
 - 30.1 给应用程序服务使用IIS 993
 - 30.2 Windows服务 993
 - 30.3 Windows服务的特性 994
 - 30.4 与Windows服务交互 995
 - 30.5 创建Windows服务 996
 - 30.5.1 用于Windows服务的.NET Framework类 996
 - 30.5.2 其他类型的Windows服务 998
 - 30.6 在VB中创建Windows服务 998
 - 30.7 创建文件监视器服务 1000
 - 30.7.1 创建Windows服务的解决方案 1000
 - 30.7.2 给服务添加.NET组件 1000
 - 30.7.3 安装服务 1003
 - 30.7.4 启动服务 1004
 - 30.7.5 卸载服务 1005
 - 30.8 与服务通信 1005
 - 30.8.1 ServiceController类 1006
 - 30.8.2 把ServiceController集成到例子中 1007
 - 30.8.3 ServiceController的

- 更多内容 1008
- 30.9 定制命令 1008
- 30.10 给服务传递字符串 1010
- 30.11 调试服务 1010
- 30.12 小结 1012
- 第31章 程序集和反射 1013
- 31.1 程序集 1013
- 31.2 清单 1014
- 31.2.1 程序集标识部分 1016
- 31.2.2 引用的程序集 1018
- 31.3 程序集与部署 1018
- 31.3.1 应用程序私有的程序集 1018
- 31.3.2 共享程序集 1019
- 31.4 版本化问题 1020
- 31.4.1 应用程序隔离 1020
- 31.4.2 并行执行 1020
- 31.4.3 自描述 1021
- 31.4.4 版本策略 1021
- 31.4.5 配置文件 1022
- 31.5 反射基础 1025
- 31.5.1 Assembly类 1026
- 31.5.2 获得当前加载的程序集 1026
- 31.5.3 Type类 1027
- 31.6 程序集的动态加载 1028
- 31.6.1 Assembly类的LoadFrom方法 1028
- 31.6.2 动态加载示例 1029
- 31.6.3 传入程序集 1030
- 31.7 小结 1031
- 第32章 .NET Framework中的安全性 1033
- 32.1 安全的概念与定义 1034
- 32.2 System.Security.Permissions名称空间中的权限 1035
- 32.2.1 代码访问权限 1037
- 32.2.2 身份权限 1038
- 32.2.3 基于角色的权限 1038
- 32.3 管理代码访问权限集合 1041
- 32.4 用户访问控制 1043
- 32.5 定义应用程序的UAC设置 1043
- 32.5.1 安全性工具 1045
- 32.5.2 使用SecurityException类处理异常 1046
- 32.6 加密基础 1047
- 32.7 小结 1060
- 第33章 使用任务和线程进行并行编程 1061
- 33.1 启动并行任务 1061
- 33.1.1 System.Threading.Tasks.Parallel类 1062
- 33.1.2 Parallel.Invoke 1062

- 33.2 把串行代码转换为并行代码 1066
 - 33.2.1 检测热点 1067
 - 33.2.2 测试并行执行获得的速度提升 1069
 - 33.2.3 理解并行和并发执行 1070
- 33.3 并行循环 1071
 - 33.3.1 Parallel.For 1071
 - 33.3.2 Parallel.ForEach 1076
 - 33.3.3 退出并行循环 1081
- 33.4 指定希望的并行度 1086
 - 33.4.1 ParallelOptions 1086
 - 33.4.2 理解硬件线程和逻辑核心 1087
- 33.5 创建和管理任务 1088
 - 33.5.1 System.Threading.Tasks.Task 1089
 - 33.5.2 理解任务的生命周期 1090
 - 33.5.3 使用任务并行化代码 1091
 - 33.5.4 从任务中返回值 1099
 - 33.5.5 为并发和并行准备代码 1102
 - 33.5.6 理解并发集合特性 1103
 - 33.5.7 把LINQ转换为PLINQ 1106
- 33.6 小结 1108
- 第34章 部署 1109
 - 34.1 应用程序部署 1110
 - 34.1.1 .NET中的部署很简单 1110
 - 34.1.2 XCOPY部署 1110
 - 34.1.3 使用Windows Installer 1110
 - 34.1.4 ClickOnce部署 1111
 - 34.2 选择Framework版本 1111
 - 34.3 Visual Studio部署项目 1112
 - 34.3.1 项目模板 1112
 - 34.3.2 创建部署项目 1113
 - 34.4 修改部署项目 1117
 - 34.4.1 项目属性 1117
 - 34.4.2 File System编辑器 1119
 - 34.4.3 Registry编辑器 1122
 - 34.4.4 File Types编辑器 1124
 - 34.4.5 User Interface编辑器 1125
 - 34.4.6 Custom Actions编辑器 1127
 - 34.4.7 Launch Conditions编辑器 1129
 - 34.4.8 构建 1132
 - 34.5 Windows应用程序的Internet部署 1132
 - 34.5.1 “无接触”部署 1132
 - 34.5.2 ClickOnce部署 1133
 - 34.6 IIS Web部署工具 1140

34.7 小结 1142
第VI部分 附录
附录 A VB编译器 1145
附录 B Visual Basic Power
Packs Tools 1161
附录 C Workflow 2008 1173
附录 D 企业服务 1193
附录 E 云的编程 1215
• • • • • ([收起](#))

[Visual Basic 2010&.NET 4高级编程_下载链接1](#)

标签

basic

.net4

visual

编程语言

编程

2010

wpf

windows

评论

[Visual Basic 2010&.NET 4高级编程_下载链接1](#)

书评

我已经忍不住了，过来再评论一次，真的是错误百出，164、172页两个实例前者运行结果不正确，后者直接出错，TMD浪费我几个小时时间调试。真得是vb.net好书难求
我已经忍不住了，过来再评论一次，真的是错误百出，164、172页两个实例前者运行结果不正确，后者直接出错，TMD浪费我几...

[Visual Basic 2010&.NET 4高级编程_下载链接1](#)