

精通C# 5.0与.NET 4.5高级编程——LINQ、WCF、WPF和WF



[精通C# 5.0与.NET 4.5高级编程——LINQ、WCF、WPF和WF_下载链接1](#)

著者:张敬普

出版者:清华大学出版社

出版时间:2014-2-14

装帧:平装

isbn:9787302333883

.net 4.5框架集成了对windows窗体、windows phone、windows server和windows azure等各个方向的开发类库。本书借助visual studio 2012开发工具，详细介绍了.net框架中的4大开发利器：linq、wpf、wf和wcf。

《精通c# 5.0与.net 4.5高级编程——linq、wcf、wpf和wf》共分为5篇28章。第1篇为c#面向对象开发基础，主要让读者了解基于框架的开发语言c#；第2篇为c#中的linq开发，介绍了如何使用linq在对象和数据之间建立一种对应关系，然后通过访问内存对象的方式查询数据集合；第3篇为wpf开发，详细介绍了如何利用wpf这个图形平台，改变传统windows程序员的开发观念和开发方式；第4篇为wf工作流开发，介绍了实现工作流应用程序时所需要实现的一些机制，比如持久化、补偿、跟踪机制等；第5篇为wcf开发，学习如何利用远程通信技术，实现托管、服务实例管理、异步、安全、事务管理、离线队列等功能。

《精通c# 5.0与.net 4.5高级编程——linq、wcf、wpf和wf》实例具有代表性，编码规范，讲解详细，适合想系统学习c#开发和.net 4.5平台的技术人员研读，也可作为特定技术开发人员的参考书。对于.net程序员，本书不失为一本极佳的案头必备手册。

作者介绍:

目录: 《精通c# 5.0与.net 4.5高级编程——linq、wcf、wpf和wf》

第1篇 c#面向对象开发基础

第1章 .net框架和visual studio 2012开发工具 2

1.1 .net 4.5框架一览 2

1.1.1 框架包含的开发语言 2

1.1.2 c#开发语言的特点 3

1.1.3 c#的运行机制 3

1.1.4 c#的类库框架 4

1.2 visual studio 2012 开发界面 4

1.2.1 visual studio 2012界面一览 4

1.2.2 菜单栏 6

1.2.3 工具栏 7

1.2.4 工具箱 7

1.2.5 属性窗口 8

1.2.6 解决方案资源管理器 8

1.2.7 服务器资源管理器 9

1.3 创建c#应用程序 9

1.3.1 控制台应用程序 (console) 9

1.3.2 windows 应用程序 (forms) 10

1.3.3 asp.net 4.5网站 11

1.4 小结 14

第2章 c#的语法知识 15

2.1 变量、常量和表达式 15

2.1.1 常量 15

2.1.2 变量 15

2.1.3 变量的类型 15

2.1.4 类型转换 16

2.1.5 变量的作用域和生存周期 17

2.1.6 表达式 17

2.2 运算符 17

2.2.1 算术运算符 18

- 2.2.2 关系运算符和逻辑运算符 18
- 2.2.3 赋值运算符 19
- 2.2.4 ? 运算符 19
- 2.2.5 运算符优先级 19
- 2.3 流程控制语句 20
 - 2.3.1 if语句 20
 - 2.3.2 switch语句 20
 - 2.3.3 for语句 21
 - 2.3.4 foreach语句 22
 - 2.3.5 while语句 22
- 2.4 方法和函数 23
 - 2.4.1 方法和函数的定义 23
 - 2.4.2 函数的参数 24
 - 2.4.3 函数的返回值 24
 - 2.4.4 函数的重载 24
 - 2.4.5 委托 25
- 2.5 泛型 25
 - 2.5.1 什么是泛型 25
 - 2.5.2 泛型的应用范围 26
- 2.6 小结 26
- 第3章 c#的面向对象特性 27
 - 3.1 面向对象编程概述 27
 - 3.1.1 什么是类和对象 27
 - 3.1.2 用c#如何创建类 28
 - 3.1.3 用c#如何创建对象 29
 - 3.1.4 什么是面向对象编程 29
 - 3.2 面向对象的技术 29
 - 3.2.1 接口 29
 - 3.2.2 用c#如何创建接口 30
 - 3.2.3 继承 31
 - 3.2.4 用c#如何实现继承 31
 - 3.2.5 多态 31
 - 3.2.6 用c#如何实现多态 32
 - 3.3 面向对象开发的简单实例 33
 - 3.3.1 面向对象功能设计 33
 - 3.3.2 面向对象的类设计 33
 - 3.3.3 面向对象的应用 34
 - 3.4 小结 35
- 第2篇 c#中的linq开发
- 第4章 linq查询基础 38
 - 4.1 linq基础概念 38
 - 4.1.1 什么是linq 38
 - 4.1.2 如何使用linq 39
 - 4.1.3 linq查询 40
 - 4.2 linq查询表达式 41
 - 4.2.1 查询表达式 41
 - 4.2.2 用from子句指定数据源 42
 - 4.2.3 用select子句指定目标数据 43
 - 4.2.4 用where子句指定筛选条件 46
 - 4.2.5 用orderby子句进行排序 48
 - 4.2.6 用group子句进行分组 49
 - 4.2.7 用from子句进行复合查询 52
 - 4.2.8 用join子句进行联接 54
 - 4.2.9 用join子句进行内部联接 54

- 4.2.10 用join子句进行分组联接 55
- 4.2.11 用join子句进行左外部联接 56
- 4.3 linq查询方法 57
 - 4.3.1 enumerable[t]接口 57
 - 4.3.2 lambda表达式 59
 - 4.3.3 用where()方法进行筛选 60
 - 4.3.4 用orderby()方法进行排序 61
 - 4.3.5 用skip()、skipwhile()跳过元素 64
 - 4.3.6 用take()、takewhile()提取元素 65
 - 4.3.7 用max()等对元素进行数值计算 66
 - 4.3.8 用distinct()消除集合中相等的元素 67
 - 4.3.9 用concat()连接两个集合 70
 - 4.3.10 用union()等进行集合操作 71
- 4.4 小结 73
- 第5章 linq to ado.net——支持 linq语言与ado.net的交互 74
 - 5.1 linq to ado.net概述 74
 - 5.1.1 linq to dataset概述 74
 - 5.1.2 linq to sql概述 75
 - 5.2 linq to dataset--实现复杂数据查询 76
 - 5.2.1 使用linq to dataset 76
 - 5.2.2 查询单个数据表 77
 - 5.2.3 查询多个数据表 80
 - 5.2.4 用查询创建数据表 83
 - 5.2.5 修改表中字段数据 85
 - 5.2.6 使用数据视图dataview 86
 - 5.3 linq to sql--对象模型 89
 - 5.3.1 对象模型的原理 89
 - 5.3.2 生成对象关系设计器 90
 - 5.3.3 对象关系设计器的构成 92
 - 5.3.4 深入linq to sql类 93
 - 5.3.5 通过绑定浏览数据库记录 95
 - 5.3.6 用linq查询linq to sql类 97
 - 5.3.7 修改数据库记录 99
 - 5.4 小结 100
- 第6章 linq to xml--支持linq语言与xml语言的交互 102
 - 6.1 linq to xml概述 102
 - 6.1.1 什么是linq to xml 102
 - 6.1.2 linq to xml与dom 104
 - 6.2 构造xml树 105
 - 6.2.1 构造简单的xml元素 105
 - 6.2.2 构造具有属性的xml元素 107
 - 6.2.3 构造具有子元素的xml元素 110
 - 6.2.4 用xelement类构造xml树 111
 - 6.3 查询xml树 112
 - 6.3.1 查找具有特定属性的元素 112
 - 6.3.2 查找具有特定子元素的元素 114
 - 6.3.3 对xml元素进行排序 115
 - 6.3.4 基于上下文查询元素 116
 - 6.4 操作xml树 117
 - 6.4.1 加载和保存xml树 118
 - 6.4.2 添加元素到xml树 119
 - 6.4.3 移除xml树的元素和属性 121
 - 6.5 小结 125
- 第3篇 wpf美轮美奂技术

第7章 wpf编程入门 128

7.1 wpf概述 128

7.1.1 理解windows图形系统 128

7.1.2 wpf的功能特点 129

7.1.3 wpf架构简介 129

7.1.4 wpf的类层次结构 130

7.2 wpf开发工具简介 132

7.2.1 visual studio 2012开发工具 132

7.2.2 microsoft expression blend设计工具 133

7.2.3 xamlpad标记文本查看器 134

7.2.4 zam 3d第三方工具 (3d模型) 135

7.3 第一个wpf应用程序 136

7.4 wpf应用程序结构 137

7.5 wpf与.net framework 4.5平台 138

7.5.1 windows communication foundation概述 138

7.5.2 windows cardspace概述 140

7.5.3 windows workflow foundation 概述 140

7.6 小结 141

第8章 xaml标记语言 143

8.1 理解xaml 143

8.1.1 xaml是什么 143

8.1.2 用c#模拟xaml实现的内容 145

8.1.3 xaml的种类 147

8.2 xaml基础 147

8.2.1 命名空间 148

8.2.2 代码后置文件 150

8.2.3 使用简单的属性和类型转换器 153

8.2.4 属性语法与属性元素语法 154

8.2.5 标记扩展特性 157

8.2.6 附加属性 157

8.2.7 特定的字符和空白 160

8.2.8 在xaml中使用事件 161

8.2.9 使用其他命名空间中的类型 163

8.2.10 加载和编译xaml 164

8.3 小结 169

第9章 application全局应用程序类 170

9.1 应用程序生命周期 170

9.1.1 创建application对象 170

9.1.2 创建一个自定义的application类 172

9.1.3 关闭应用程序 173

9.1.4 应用程序事件 174

9.2 事件执行周期 175

9.2.1 startup应用程序启动事件 175

9.2.2 activated和deactivated事件——激活与取消激活 176

9.2.3 dispatcherunhandledexception事件——处理应用程序未处理的异常 177

9.2.4 sessionending事件——注销或关闭系统 178

9.2.5 exit事件——退出应用程序 179

9.3 application类的任务 181

9.3.1 处理命令行参数 181

9.3.2 访问当前的应用程序 182

9.3.3 单实例应用程序 185

9.4 小结 190

第10章 wpf布局 191

10.1 理解wpf布局 191

- 10.1.1 wpf的布局原理 191
- 10.1.2 布局的过程 192
- 10.1.3 布局容器 192
- 10.2 使用stackpanel进行简单地布局 193
 - 10.2.1 stackpanel示例 193
 - 10.2.2 stackpanel布局属性 195
- 10.3 使用wrappanel和dockpanel 197
 - 10.3.1 使用wrappanel控件布局 197
 - 10.3.2 使用dockpanel控件布局 198
 - 10.3.3 简单对话框的实现 200
- 10.4 grid控件 201
 - 10.4.1 创建grid控件 201
 - 10.4.2 调整行列尺寸 203
 - 10.4.3 编程创建grid控件 206
 - 10.4.4 合并行和列 208
 - 10.4.5 窗体分割 209
 - 10.4.6 共享尺寸组 212
 - 10.4.7 使用uniformgrid控件 216
- 10.5 基于坐标布局的canvas 217
 - 10.5.1 canvas控件的使用方法 217
 - 10.5.2 canvas的z轴 218
 - 10.5.3 inkcanvas介绍 219
- 10.6 小结 221
- 第11章 内容控件 222
 - 11.1 理解内容控件 222
 - 11.1.1 什么是内容控件 222
 - 11.1.2 content属性介绍 224
 - 11.1.3 内容的对齐属性 226
 - 11.2 内容容器控件 227
 - 11.2.1 scrollviewer滚动条控件 227
 - 11.2.2 groupbox组合框和tabitem标签页控件 231
 - 11.2.3 expander可折叠控件 234
 - 11.3 装饰控件 236
 - 11.3.1 border边框控件 237
 - 11.3.2 viewbox自动缩放控件 238
 - 11.4 小结 239
- 第12章 wpf依赖属性和事件路由 240
 - 12.1 wpf中的树 240
 - 12.1.1 逻辑树和视觉树 240
 - 12.1.2 编程遍历树结构 242
 - 12.2 理解依赖属性 244
 - 12.2.1 依赖属性的概念 244
 - 12.2.2 定义依赖属性 246
 - 12.2.3 依赖属性示例 249
 - 12.2.4 共享依赖属性 253
 - 12.2.5 注册附加属性 254
 - 12.3 理解事件路由 255
 - 12.3.1 路由事件的概念 255
 - 12.3.2 定义和注册路由事件 257
 - 12.3.3 事件路由策略 259
 - 12.3.4 routedeventargs类型的参数 262
 - 12.3.5 附加事件 263
 - 12.4 wpf中的事件 264
 - 12.4.1 生命周期事件 264

- 12.4.2 键盘事件 265
- 12.4.3 键盘焦点 269
- 12.4.4 键盘状态 270
- 12.4.5 鼠标事件 271
- 12.4.6 捕捉鼠标 275
- 12.4.7 鼠标拖放编程 277
- 12.4.8 文件拖放示例 279
- 12.5 小结 282
- 第13章 标准控件 283
 - 13.1 控件基础 283
 - 13.1.1 背景色和前景色画刷 283
 - 13.1.2 透明度设置 285
 - 13.1.3 更改字体 286
 - 13.1.4 更改鼠标光标 287
 - 13.2 内容控件 291
 - 13.2.1 label标签控件 291
 - 13.2.2 button控件 292
 - 13.2.3 checkbox和radiobutton控件 295
 - 13.2.4 tooltip提示控件 296
 - 13.2.5 使用tooltipservice对象tooltip 298
 - 13.2.6 使用popup控件 300
 - 13.3 文本控件 303
 - 13.3.1 textbox控件 304
 - 13.3.2 选择文本 305
 - 13.3.3 passwordbox密码框控件 306
 - 13.3.4 textblock控件 308
 - 13.4 列表控件 310
 - 13.4.1 listbox列表框控件 310
 - 13.4.2 获取listbox选定项 312
 - 13.4.3 在listbox中进行多选 314
 - 13.4.4 combobox下拉列表框控件 315
 - 13.5 范围控件 318
 - 13.5.1 progressbar进度条控件 318
 - 13.5.2 slider滑动条控件 319
 - 13.6 小结 322
- 第14章 窗口与导航 323
 - 14.1 windows类简介 323
 - 14.1.1 window类的基本属性 323
 - 14.1.2 窗口的生命周期 326
 - 14.1.3 模式与非模式窗口 329
 - 14.1.4 窗口的定位和尺寸 330
 - 14.1.5 保存和恢复窗口位置 331
 - 14.2 对话框窗口 332
 - 14.2.1 窗口的宿主 332
 - 14.2.2 对话框模型 333
 - 14.2.3 messagebox消息框 335
 - 14.2.4 通用对话框 336
 - 14.3 非矩形窗口 338
 - 14.3.1 圆角窗口 339
 - 14.3.2 图形窗口 340
 - 14.3.3 调整非矩形窗口尺寸 342
 - 14.4 一个简单的导航应用程序示例 343
 - 14.5 基于页面的用户界面 345
 - 14.5.1 page类简介 345

- 14.5.2 使用hyperlink超级链接 346
- 14.5.3 片段导航 347
- 14.5.4 navigationwindow导航窗口 348
- 14.5.5 frame页面框架 349
- 14.5.6 导航历史记录 352
- 14.6 navigationservice导航服务 353
 - 14.6.1 编程控制导航 353
 - 14.6.2 导航生命期事件 355
 - 14.6.3 导航记录管理 358
 - 14.6.4 使用页函数pagefunction 363
- 14.7 xaml浏览器应用程序 366
 - 14.7.1 xbap的特点 367
 - 14.7.2 创建xbap应用程序 367
- 14.8 小结 368
- 第15章 wpf资源、样式和控件模板 369
 - 15.1 程序集资源 369
 - 15.1.1 添加资源 369
 - 15.1.2 获取程序集资源 369
 - 15.2 对象资源 371
 - 15.2.1 资源集合 372
 - 15.2.2 静态和动态资源 373
 - 15.2.3 非共享资源 374
 - 15.2.4 编程访问资源 375
 - 15.2.5 应用程序资源 375
 - 15.2.6 系统资源 376
 - 15.2.7 使用资源字典组织资源 377
 - 15.3 样式基础 378
 - 15.3.1 创建一个样式对象 379
 - 15.3.2 在样式中设置属性 380
 - 15.3.3 根据指定的类型自动应用样式 381
 - 15.3.4 多级样式 382
 - 15.3.5 绑定事件处理器 382
 - 15.4 样式触发器 384
 - 15.4.1 创建简单触发器 384
 - 15.4.2 创建事件触发器 387
 - 15.4.3 数据触发器 387
 - 15.5 控件模板 390
 - 15.5.1 理解控件模板 390
 - 15.5.2 一个按钮模板示例 390
 - 15.5.3 模板的绑定 392
 - 15.5.4 模板触发器 393
 - 15.6 小结 395
- 第16章 形状、变换和画刷 396
 - 16.1 wpf图形 396
 - 16.1.1 shape类 396
 - 16.1.2 rectangle和ellipse类 397
 - 16.1.3 stretch属性 398
 - 16.1.4 line直线对象 399
 - 16.1.5 polyline多段线对象 400
 - 16.1.6 polygon多边形对象 401
 - 16.1.7 线帽和线连接点 403
 - 16.1.8 绘制虚线 404
 - 16.2 画刷 405
 - 16.2.1 lineargradientbrush线性渐变画刷 405

- 16.2.2 radialgradientbrush径向渐变画刷 407
- 16.2.3 imagebrush图像画刷 409
- 16.2.4 visualbrush可视化对象画刷 410
- 16.2.5 opacitymask不透明蒙板 411
- 16.3 wpf变换 414
 - 16.3.1 应用变换 414
 - 16.3.2 translatetransform移动变换 415
 - 16.3.3 rotatetransform旋转变换 416
 - 16.3.4 scaletransform缩放变换 417
 - 16.3.5 skewtransform扭曲变换 418
 - 16.3.6 transformgroup组合变换 419
- 16.4 位图效果 420
 - 16.4.1 模糊 420
 - 16.4.2 凹凸效果 422
 - 16.4.3 浮雕效果 423
 - 16.4.4 发光和阴影 424
- 16.5 小结 425
- 第17章 几何图形、图像和可视化层 426
 - 17.1 路径和几何图形 426
 - 17.1.1 线型、矩形和椭圆几何图形 426
 - 17.1.2 使用geometrygroup组合形状 427
 - 17.1.3 使用combinedgeometry结合形状 428
 - 17.1.4 认识pathgeometry对象 430
 - 17.1.5 用pathgeometry对象绘制直线 431
 - 17.1.6 用pathgeometry对象绘制弧线 431
 - 17.1.7 用pathgeometry对象绘制贝塞尔曲线 432
 - 17.1.8 使用几何迷你语言 433
 - 17.1.9 几何图形的裁切 434
 - 17.2 绘图 435
 - 17.2.1 绘制形状 436
 - 17.2.2 绘制图像 437
 - 17.2.3 组合绘制 438
 - 17.3 可视化层 439
 - 17.3.1 在visual上绘图 439
 - 17.3.2 drawingvisual宿主容器 441
 - 17.3.3 使用命中测试 443
 - 17.4 小结 445
- 第18章 wpf数据绑定 446
 - 18.1 数据绑定基础 446
 - 18.1.1 绑定到元素属性 446
 - 18.1.2 使用程序代码创建绑定 447
 - 18.1.3 绑定多个属性 448
 - 18.1.4 绑定的方向 449
 - 18.1.5 绑定更新 450
 - 18.1.6 绑定到非元素的对象 452
 - 18.2 数据库绑定 456
 - 18.2.1 创建数据访问组件 456
 - 18.2.2 实现数据实体对象 458
 - 18.2.3 显示绑定对象 459
 - 18.2.4 更新数据库 461
 - 18.3 小结 463
- 第4篇 wf工作流编程
- 第19章 wf编程入门 466
 - 19.1 wf基础 466

- 19.1.1 wf简介 466
- 19.1.2 wf的开发环境 468
- 19.1.3 第一个工作流示例程序 469
- 19.1.4 wf架构简介 473
- 19.2 使用workflowruntime和workflowinstance类 474
- 19.2.1 理解workflowruntime工作流引擎 475
- 19.2.2 理解workflowinstance工作流实例 478
- 19.3 工作流类型和创建模式 479
- 19.3.1 wf工作流的类型 480
- 19.3.2 一个简单的状态机工作流示例 480
- 19.3.3 wf工作流的创建模式 483
- 19.4 小结 484
- 第20章 wf活动 485
- 20.1 理解wf活动 485
- 20.1.1 活动的类层次结构 485
- 20.1.2 活动条件类型 485
- 20.2 使用wf的基本活动 488
- 20.2.1 使用ifelseactivity活动 489
- 20.2.2 使用whileactivity活动 490
- 20.2.3 使用paralleactivity活动 491
- 20.2.4 使用replicatoractivity活动 494
- 20.2.5 使用conditionedactivitygroup活动 498
- 20.2.6 使用invokeworkflowactivity活动 501
- 20.2.7 使用terminateactivity活动 504
- 20.2.8 使用suspendactivity活动 504
- 20.3 本地服务和事件驱动的活动 507
- 20.3.1 理解和实现本地服务 507
- 20.3.2 使用本地服务在宿主和工作流之间通信 508
- 20.3.3 事件驱动的活动简介 512
- 20.3.4 eventdrivenactivity和listenactivity使用示例 513
- 20.3.5 使用eventhandlingscopeactivity活动 516
- 20.4 小结 522
- 第21章 wf服务 523
- 21.1 wf服务基础 523
- 21.1.1 wf服务的分类 523
- 21.1.2 默认的服务类简介 524
- 21.2 wf持久化服务 525
- 21.2.1 准备数据库 525
- 21.2.2 创建工作流 527
- 21.2.3 使用sqlworkflowpersistenceservice 528
- 21.3 wf跟踪服务 536
- 21.3.1 使用工作流跟踪服务 536
- 21.3.2 跟踪服务配置文件 540
- 21.4 小结 541
- 第22章 基于asp.net的工作流批核系统 542
- 22.1 工作流批核系统简介 542
- 22.1.1 系统运行效果 543
- 22.1.2 系统基本结构 546
- 22.2 workflows工作流项目 547
- 22.2.1 基于事件驱动的活动实现 548
- 22.2.2 useractivity用户活动实现 549
- 22.2.3 saveworkitems自定义活动实现 556
- 22.2.4 approvedworkitemworkflow工作流实例的实现 557
- 22.2.5 工作流项目帮助类的实现 559

- 22.3 asp.net宿主应用程序项目 561
 - 22.3.1 初始化工作流运行时引擎 561
 - 22.3.2 创建工作流实例 562
 - 22.3.3 编辑和审核工作申请表单 564
- 22.4 小结 568
- 第5篇 wcf开发
- 第23章 soa和wcf基础介绍 570
 - 23.1 soa架构介绍 570
 - 23.1.1 软件设计思想发展的简要介绍 570
 - 23.1.2 什么是soa 571
 - 23.2 wcf简要介绍 572
 - 23.2.1 什么是wcf 572
 - 23.2.2 wcf体系框架 573
 - 23.2.3 wcf基础概念介绍 575
 - 23.3 第一个wcf程序 577
 - 23.3.1 helloworld服务契约的定义 577
 - 23.3.2 helloworld的宿主程序 579
 - 23.3.3 访问helloworld服务的客户端程序 581
 - 23.4 小结 583
- 第24章 通道模型和绑定 584
 - 24.1 wcf通道模型 584
 - 24.1.1 wcf通道模型概述 584
 - 24.1.2 消息交换模式和通道形状 585
 - 24.1.3 数据报模式 585
 - 24.1.4 请求-响应模式 588
 - 24.1.5 双工模式 590
 - 24.1.6 带会话的数据报模式、请求-响应模式和双工模式 592
 - 24.1.7 通道形状的改变 592
 - 24.1.8 通道形状和上层服务协议 593
 - 24.1.9 通道管理器 594
 - 24.1.10 icomcommunicationobject接口和状态改变 595
 - 24.2 标准绑定介绍 598
 - 24.2.1 绑定的基本概念 598
 - 24.2.2 标准绑定 599
 - 24.2.3 设置绑定的方式 600
 - 24.2.4 如何选择绑定 601
 - 24.3 本机wcf-wcf交互的绑定和地址 602
 - 24.3.1 场景概述 602
 - 24.3.2 ipc基本概念 603
 - 24.3.3 使用netnamedpipebinding 604
 - 24.3.4 netnamedpipebinding的地址和配置 605
 - 24.3.5 netnamedpipebinding特点总结 607
 - 24.4 跨主机wcf-wcf交互的绑定和地址 607
 - 24.4.1 场景概述 607
 - 24.4.2 tcp协议概述 607
 - 24.4.3 net.tcp端口共享 608
 - 24.4.4 使用nettcpbinding 609
 - 24.4.5 nettcpbinding的地址和配置 610
 - 24.4.6 nettcpbinding特点总结 612
 - 24.5 与ws-i basic web服务进行交互的绑定和地址 613
 - 24.5.1 场景概述 613
 - 24.5.2 soap协议概述 613
 - 24.5.3 使用basichttpbinding 614
 - 24.5.4 basichttpbinding的地址和配置 615

- 24.5.5 basichttpbinding特点总结 617
- 24.6 与改进web服务进行交互的绑定和地址 618
 - 24.6.1 场景概述 618
 - 24.6.2 改进web服务协议概述 618
 - 24.6.3 使用wshttpbinding 622
 - 24.6.4 wshttpbinding的地址和配置 624
 - 24.6.5 wshttpbinding特点总结 627
 - 24.6.6 使用wsdualhttpbinding 627
 - 24.6.7 wsdualhttpbinding的地址和配置 628
 - 24.6.8 wsdualhttpbinding特点总结 630
 - 24.6.9 使用ws2007httpbinding 630
- 24.7 使用脱机模式进行消息交互的绑定和地址 631
 - 24.7.1 场景概述 631
 - 24.7.2 msmq协议概述 631
 - 24.7.3 使用netmsmqbinding 633
 - 24.7.4 netmsmqbinding的地址和配置 634
 - 24.7.5 netmsmqbinding特点总结 636
 - 24.7.6 使用msmqintegrationbinding 636
 - 24.7.7 msmqintegrationbinding的地址和配置 637
 - 24.7.8 msmqintegrationbinding特点总结 639
- 24.8 小结 640

第25章 定义服务：契约编程 641

- 25.1 契约定义和分类 641
 - 25.1.1 什么是契约 641
 - 25.1.2 契约分类 641
- 25.2 服务契约 642
 - 25.2.1 服务契约和wsdl 642
 - 25.2.2 服务契约的重载问题 644
 - 25.2.3 定义请求-响应操作 646
 - 25.2.4 定义单程操作 647
 - 25.2.5 定义双程操作 649
 - 25.2.6 wcf中事件的实现 657
- 25.3 数据契约 661
 - 25.3.1 数据契约和xsd 661
 - 25.3.2 使用datacontract特性定义数据契约 663
 - 25.3.3 数据契约的继承 666
 - 25.3.4 已知类型的定义 667
 - 25.3.5 数据契约的等效性 672
 - 25.3.6 数据契约的版本控制 675
 - 25.3.7 定义必需的数据成员 678
 - 25.3.8 数据默认值的发送 680
 - 25.3.9 数组和集合的处理 680
- 25.4 消息契约 684
 - 25.4.1 消息契约基本概念 684
 - 25.4.2 强类型消息 686
 - 25.4.3 弱类型消息 692
- 25.5 错误处理和错误契约 695
 - 25.5.1 soap消息的错误处理 695
 - 25.5.2 服务端未捕获的异常 696
 - 25.5.3 包含详细异常信息 698
 - 25.5.4 捕捉服务异常 699
 - 25.5.5 faultcode和faultreason的使用 701
 - 25.5.6 使用错误契约和faultexception[t] 703
- 25.6 小结 706

- 第26章 wcf中的行为 707
 - 26.1 实例管理 707
 - 26.1.1 实例管理的设置 707
 - 26.1.2 percall实例策略 708
 - 26.1.3 persession实例策略和会话 710
 - 26.1.4 single实例策略 713
 - 26.2 并发管理 717
 - 26.2.1 并发管理的设置 717
 - 26.2.2 single并发模式 717
 - 26.2.3 multiple并发模式 720
 - 26.2.4 reentrant并发模式 721
 - 26.3 元数据的发布 724
 - 26.3.1 http-get方式发布元数据 724
 - 26.3.2 mex终结点方式发布元数据 727
 - 26.4 事务管理 728
 - 26.4.1 事务的基本概念和特性 728
 - 26.4.2 单服务事务 729
 - 26.4.3 分布式事务 733
 - 26.4.4 事务协议和事务管理器 739
 - 26.5 小结 741
- 第27章 安全 742
 - 27.1 常用概念介绍 742
 - 27.1.1 身份验证 742
 - 27.1.2 授权 743
 - 27.1.3 保密性 743
 - 27.1.4 完整性 743
 - 27.1.5 凭据 743
 - 27.2 传输安全性 744
 - 27.2.1 传输安全性的三要素 744
 - 27.2.2 wcf支持的传输安全模式 744
 - 27.2.3 传输安全模式的配置 745
 - 27.2.4 transport安全模式下的凭证 746
 - 27.2.5 message安全模式下的凭证 747
 - 27.3 局域网内windows平台系统的安全性 747
 - 27.3.1 场景分析 747
 - 27.3.2 服务定义 748
 - 27.3.3 身份认证 750
 - 27.3.4 授权 752
 - 27.4 跨internet系统的安全性 756
 - 27.4.1 场景概述和绑定选择 756
 - 27.4.2 消息安全 757
 - 27.4.3 身份认证 760
 - 27.4.4 授权 762
 - 27.5 安全机制的日志和跟踪 763
 - 27.6 小结 764
- 第28章 自动点滴管理系统 765
 - 28.1 需求分析 765
 - 28.1.1 系统介绍 765
 - 28.1.2 客户端功能 765
 - 28.1.3 服务端功能 766
 - 28.2 系统设计 766
 - 28.2.1 服务边界和接口 766
 - 28.2.2 传输和寄宿的设计 766
 - 28.2.3 界面的设计 767

- 28.3 系统实现 768
 - 28.3.1 服务契约的定义 768
 - 28.3.2 数据契约的定义 769
 - 28.3.3 服务的实现 774
 - 28.3.4 控制台实现 780
 - 28.3.5 客户端实现 788
- 28.4 运行和测试 796
- 28.5 小结 797
- • • • • [\(收起\)](#)

[精通C# 5.0与.NET 4.5高级编程——LINQ、WCF、WPF和WF_下载链接1](#)

标签

.net

编程

计算机

C

#C

#

评论

[精通C# 5.0与.NET 4.5高级编程——LINQ、WCF、WPF和WF_下载链接1](#)

书评

[精通C# 5.0与.NET 4.5高级编程——LINQ、WCF、WPF和WF_下载链接1](#)