

《C#与.NET 3.5高级程序设计》

图书基本信息

书名：《C#与.NET 3.5高级程序设计》

13位ISBN编号：9787115196910

10位ISBN编号：7115196915

出版时间：2009-3

出版社：人民邮电出版社

作者：特罗尔森

页数：1107

译者：朱晔,肖逵,张大磊,王少葵,范睿

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu111.com

前言

~应用程序开发的目标始终如一，就是在最短时间内制作出最好的软件。当今最流行的快速开发平台是什么，毫无疑问，是微软公司的.NET Framework。经过近8年的发展，微软的.NET已经变得非常庞大和成熟，而且.NET发展的步伐越来越快。2002年年初，微软发布了Visual Studio.NET 2002开发工具，也带来了.NET框架的第一个版本1.0。2003年春天，微软发布了Visual Studio.NET 2003以及.NET Framework 1.1。尽管只是1.1，但是.NET Framework 1.1比1.0多了几项重大更新，例如对移动设备的支持（精简版.NET）和对ODBC / Oracle数据库的支持。2005年年底，.NET Framework 2.0随着Visual Studio 2005以及SQL Server 2005一起发布。无论是Visual Studio 2005、C#2.0还是ASP.NET 2.0，在易用性、安全性等方面都比前辈优秀很多。C#2.0中的泛型、迭代器、可空类型、匿名方法、分部类等特性确实给了我们很多方便。2006年年底，微软随Vista操作系统推出了.NET Framework 3.0。.NET Framework 3.0在2.0的基础上增加了几个重要组件：WCF、WPF、WF和WCS（本书后面会对WCF、WPF、WF、WCS进行比较详细的介绍）。WCF：Windows Communication Foundation，最完整的集成化通信管理框架。WCF编程模型把Web服务、.NET远程技术、分布式事务和消息队列统一到单个面向服务的编程模型中，从而实现了真正意义上的分布式计算。WPF：Windows Presentation Foundation，最华丽的界面技术框架。WPF提供了一种一致的方案来构建编程模型，并且支持使用更为丰富的控件和设计技术来开发Windows程序。WF：Windows Workflow Foundation，最灵活和最强大的工作流平台。WF是微软用来定义、执行和管理工作流的编程模型、引擎和工具的总称。WCS：Windows CardSpace，最安全的个人数字标识解决方案。2007年年底，微软发布了Visual Studio 2008以及.NET Framework 3.5。在.NET Framework 3.0的基础上，3.5又新增了一些程序集，并引入了LINQ、ASP.NET AJAX等功能（本书基于C#3.0，而且也会对LINQ技术进行简单介绍）。C#3.0：提供了隐式类型变量、自动属性、扩展方法、对象初始化语法、Lambda表达式、匿名方法、分部方法等新特性，其实几乎所有的特性都是因LINQ而生的。LINQ：语言集成查询，官方组件包括LINQ to SQL、LINQ to XML、LINQ to DataSet等。当然，还有很多第三方公司或个人推出的LINQ to XXX。可以通过这些组件使用统一的查询语言对各种基于磁盘、网络甚至服务的数据源进行查询。ASP.NET AJAX：其实ASP.NET AJAX最早并不是在.NET Framework 3.5中出现的，只不过.NET Framework 3.5集成了ASP.NET AJAX。关于.NET Framework 3.0中四大技术的扩展，我们也可以认为.NET Framework 3.5提供了WCF / WPF / WF / WCS的2.0版本。~

《C#与.NET 3.5高级程序设计》

内容概要

《C#与.NET 3.5高级程序设计》(第4版)是C#领域久负盛名的经典著作，深入全面地叙述了C#编程语言和.NET平台核心，并以大量示例剖析相关概念。书中介绍了C#的各种语言构造、.NET 2.0的类、核心API、公共中间语言(CIL)、动态程序集和ASP.NET扩展等内容；同时也介绍了.NET 3.0和.NET 3.5中新的编程API，包括WPF、WCF和WF的功能；另外，还介绍了最新的C# 3.0编程语言、LINQ编程技术、COM与.NET的互操作性以及平台无关的.NET开发。

《C#与.NET 3.5高级程序设计》

作者简介

Lars Powers 拥有15年从业经验，目前是微软公司开发人员与平台推广的ISV（独立软件开发商）技术指导，负责协助微软最大的全球ISV合作伙伴，基于微软下一代技术创建解决方案。加入微软之前，他是一名独立咨询师，提供关于.NET平台的培训和指导。

Mike Snell 微软区域经理，软件架构师、顾问、.NET导师，CEI公司负责解决方案交付的工程副总裁。拥有丰富的企业级开发和商业软件开发经验，先后指导过数千名开发者。曾和Lars Powers共同编著了Visual Basic Programmer's Guide to the .NET Framework Class Library一书。

书籍目录

第一部分 visual studio 2008简介.

第1章 visual studio 2008快速入门

2

1.1 ide中一些受欢迎的增强

3

1.1.1 在多项工作中使用一个工具

3

1.1.2 更整齐的窗口

4

1.1.3 保留设置

7

1.1.4 共享（和使用）社区代码

8

1.1.5 扩展的class designer支持

8

1.2 开发用户应用程序

8

1.2.1 提高web开发人员生产力

9

1.2.2 更智能的客户端

17

1.2.3 基于office的解决方案

20

1.2.4 以移动设备为目标

22

1.3 编写连接的、面向服务的解决方案

24

1.3.1 开发应用程序/商业流程

24

1.3.2 创建和使用服务

26

1.4 使用数据

27

1.4.1 设计数据

28

1.4.2 将对象映射到关系数据

28

1.4.3 构建偶尔连接的应用程序

29

1.5 visual studio产品系列

31

1.5.1 速成版

31

1.5.2 标准版

31

1.5.3 专业版

32

1.5.4 team systems

32

1.5.5 expression工具

34

1.6 小结

34

第2章 ide快速入门

35

2.1 安装visual studio

35

2.1.1 选择语言

35

2.1.2 配置开发环境

36

2.2 起始页

37

2.3 第一个项目

39

2.4 菜单栏

39

2.5 各种工具栏

42

2.5.1 标准工具栏

42

2.5.2 自定义工具栏

43

2.6 工具箱

44

2.7 可视化设计器

45

2.8 文本编辑器

46

2.8.1 代码编辑器

46

2.8.2 编辑器的自定义

47

2.9 解决方案资源管理器

48

2.10 属性窗口

49

2.11 管理ide中的各种窗口

49

2.11.1 钉住

50

2.11.2 停靠

50

2.12 小结

52

第3章 visual studio 2008中的.netframework和语言增强

53	
3.1	从.net语言的角度概览visual studio 2008 ide增强
53	
3.1.1	vb ide增强
54	
3.1.2	c# ide增强
54	
3.2	visual studio 2008中的.net语言扩展
55	
3.2.1	根据赋值推断变量数据类型
56	
3.2.2	用一行代码创建对象并设置属性
57	
3.2.3	给已有类添加方法
58	
3.2.4	创建不存在的类的实例
59	
3.2.5	在代码中编写简单的未命名的函数
60	
3.2.6	为生成的代码添加业务逻辑
61	
3.2.7	使用.net语言访问和查询数据
63	
3.2.8	将程序集拆分为多个文件
64	
3.2.9	在代码中直接处理xml
64	
3.2.10	从事件处理程序移除未使用的参数（仅vb）
65	
3.2.11	创建自动实现的属性（仅c#）
65	
3.3	.net framework 3.5增强
66	
3.4	小结
68	
第二部分 深入ide	
第4章 解决方案和项目	
70	
4.1	解决方案
70	
4.1.1	创建解决方案
70	
4.1.2	使用解决方案
74	
4.2	项目
78	
4.2.1	创建项目
79	
4.2.2	使用项目定义文件

81	
4.2.3	使用项目
86	
4.3	小结
89	
第5章	浏览器和资源管理器
91	
5.1	解决方案资源管理器
91	
5.1.1	提示文字和图标
92	
5.1.2	管理解决方案
95	
5.1.3	管理项目
96	
5.2	类视图
96	
5.2.1	工具栏
97	
5.2.2	搜索栏
97	
5.2.3	对象窗格
97	
5.2.4	成员窗格
98	
5.3	服务器资源管理器
99	
5.3.1	数据连接
100	
5.3.2	服务器组件
100	
5.4	对象浏览器
102	
5.4.1	改变范围
103	
5.4.2	浏览对象
104	
5.5	文档大纲
105	
5.6	小结
106	
第6章	编辑器和设计器简介
107	
6.1	基础知识
107	
6.1.1	文本编辑器
107	
6.1.2	visual studio设计器
109	

6.2	使用代码编辑器编写代码	109
6.2.1	打开编辑器	110
6.2.2	在代码编辑器中编写代码	110
6.2.3	剖析代码编辑器窗口	111
6.2.4	代码导航工具	113
6.2.5	搜索文档	114
6.2.6	在文本编辑器中调试	119
6.2.7	打印代码	121
6.2.8	使用代码定义窗口	122
6.3	创建和编辑xml文档和模式	123
6.3.1	推断模式	123
6.3.2	编辑xslt样式表	124
6.4	使用css	124
6.4.1	添加样式规则	124
6.4.2	定义样式表属性	125
6.5	开发windows客户端应用程序	125
6.5.1	创建windows forms项目	125
6.5.2	创建一个wpf项目	131
6.6	开发web窗体	132
6.7	授权组件和控件	136
6.7.1	创建新的组件或控件	136
6.7.2	编写组件代码的更多说明	137
6.8	小结	138
第7章	.net社区——使用和创建共享代码	139
7.1	visual studio的社区功能	

139	
7.1.1	visual studio起始页
140	
7.1.2	帮助菜单
143	
7.2	发现和使用共享内容
151	
7.2.1	共享内容的类型
151	
7.2.2	查找正确的内容
152	
7.2.3	安装和保存共享内容
153	
7.3	回馈社区
153	
7.3.1	创建共享项（项目和项模板）
154	
7.3.2	创建项目模板
154	
7.3.3	创建项模板
158	
7.3.4	将作品打包
159	
7.3.5	发布作品
164	
7.4	小结
164	
	第三部分 编写和使用代码
	第8章 visual studio中的生产力工具
166	
8.1	文本编辑器的基本辅助功能
167	
8.1.1	变更跟踪
167	
8.1.2	代码问题指示器
168	
8.1.3	活动超链接
168	
8.1.4	语法着色
168	
8.2	大纲和导航
169	
8.2.1	代码大纲
169	
8.2.2	html导航
171	
8.3	智能标签和智能任务
172	
8.3.1	html设计器

172	
8.3.2	windows forms设计器
172	
8.3.3	代码编辑器
173	
8.4	智能感知
174	
8.4.1	完成单词
174	
8.4.2	快速信息
175	
8.4.3	列表成员
175	
8.4.4	参数信息
176	
8.4.5	管理using语句
177	
8.4.6	代码段和模板代码
177	
8.4.7	花括号匹配
184	
8.4.8	自定义智能感知
185	
8.5	任务列表
186	
8.5.1	注释任务
186	
8.5.2	快捷任务
187	
8.5.3	用户任务
187	
8.6	小结
188	
第9章	重构代码
189	
9.1	visual studio重构基础
190	
9.1.1	调用重构工具
190	
9.1.2	预览更改
193	
9.2	重命名
194	
9.2.1	访问重命名操作
194	
9.2.2	使用重命名对话框
195	
9.3	提取方法
196	

9.3.1	执行提取方法操作	196
9.3.2	提取方法	197
9.3.3	生成方法存根	202
9.4	提取接口	202
9.4.1	执行提取接口操作	203
9.4.2	提取接口	203
9.5	重构参数	205
9.5.1	移除参数	205
9.5.2	将局部变量提升为参数	206
9.5.3	重新排列参数	207
9.6	封装字段	208
9.6.1	执行封装字段操作	208
9.6.2	封装字段对话框	208
9.7	小结	209
第10章	代码调试	210
10.1	调试基础	210
10.1.1	场景	211
10.1.2	调试的多个阶段	211
10.1.3	调试应用程序（自检查）	211
10.1.4	调试基础小结	218
10.2	visual studio调试器	219
10.2.1	调试菜单和工具栏	219
10.2.2	调试选项	223
10.2.3	逐语句、跳出和逐过程代码调试	223
10.2.4	在代码中指定断点	

228	
10.2.5	使用跟踪点
233	
10.2.6	在调试器中查看数据
235	
10.2.7	使用编辑并继续特性
238	
10.3	高级调试场景
239	
10.3.1	远程调试
239	
10.3.2	调试wcf服务
240	
10.3.3	调试多线程应用程序
240	
10.3.4	调试客户端脚本
243	
10.4	小结
243	
第四部分 扩展visual studio	
第11章 自动化对象模型介绍	
246	
11.1	自动化对象模型概述
246	
11.1.1	对象模型版本
247	
11.1.2	自动化的种类
248	
11.1.3	dte/dte2根对象
248	
11.2	solution和project对象
249	
11.2.1	控制解决方案中的项目
251	
11.2.2	访问项目中的代码
252	
11.3	窗口
255	
11.3.1	引用窗口
255	
11.3.2	与窗口交互
255	
11.3.3	文本窗口和窗格
258	
11.3.4	工具窗口的类型
259	
11.3.5	链接窗口
265	
11.4	命令栏

267	
11.5	文档
269	
11.6	命令对象
279	
11.6.1	执行命令
280	
11.6.2	映射快捷键
281	
11.7	调试器对象
281	
11.8	自动化事件
282	
11.9	小结
282	
第12章	编写宏
284	
12.1	记录宏
284	
12.2	使用宏资源管理器
285	
12.3	使用宏ide编写宏
287	
12.3.1	使用宏项目
287	
12.3.2	编写宏
290	
12.3.3	调试
292	
12.3.4	处理事件
292	
12.3.5	调用宏
297	
12.4	小结
299	
第13章	编写外接程序和向导
300	
13.1	创建第一个外接程序项目
300	
13.2	外接程序的结构
307	
13.2.1	外接程序的生命周期
308	
13.2.2	响应命令
312	
13.2.3	管理外接程序
313	
13.3	示例外接程序——调色板..
314	

13.3.1	让我们开始吧	315
13.3.2	创建用户控件	315
13.3.3	完成connect类	318
13.3.4	公开外接程序的设置	320
13.4	创建visual studio向导	333
13.4.1	向导的结构	333
13.4.2	创建add new item向导	336
13.5	小结	340
第五部分 创建企业应用程序		
第14章 创建asp.net应用程序		
14.1	asp.net网站基础	342
14.1.1	新建web应用程序项目	343
14.1.2	控制项目属性和选项	352
14.1.3	创建网页	356
14.2	设计用户界面	363
14.2.1	确定页面布局和控件位置	364
14.2.2	创建通用的界面外观	367
14.2.3	创建用户可配置的ui	381
14.3	使用asp.net控件	388
14.3.1	asp.net控件概述	389
14.3.2	asp.net标准控件	390
14.3.3	验证控件	391
14.3.4	登录控件	393
14.3.5	网站导航控件	395
14.3.6	data控件	396

14.3.7	用户控件	
397		
14.4	小结	
398		
第15章	构建windows forms应用程序	
399		
15.1	窗体设计基础	
399		
15.1.1	考虑最终用户	
399		
15.1.2	ui标准的角色	
400		
15.1.3	规划用户界面	
401		
15.2	创建窗体	
401		
15.2.1	windows应用程序项目类型	
402		
15.2.2	窗体的属性和事件	
403		
15.3	添加控件和组件	
404		
15.3.1	控件的布局和定位	
405		
15.3.2	使用容器	
407		
15.3.3	控件的外观和行为	
410		
15.3.4	使用toolstrip控件	
411		
15.3.5	显示数据	
416		
15.4	创建自定义控件	
420		
15.4.1	子类化现有控件	
420		
15.4.2	设计用户控件	
421		
15.4.3	创建自定义控件	
422		
15.5	小结	
423		
第16章	创建丰富、智能的用户界面	
424		
16.1	wpf平台	
424		
16.2	wpf设计器介绍	
427		
16.3	wpf编程	

431	
16.3.1	布局
431	
16.3.2	样式和模板
435	
16.3.3	数据绑定
437	
16.3.4	路由事件
438	
16.4	创建简单的图像查看应用程序
439	
16.4.1	开始布局
440	
16.4.2	保存图像
441	
16.4.3	绑定图像
443	
16.4.4	按钮事件处理程序和图像效果
444	
16.4.5	使用通用对话框选择路径
445	
16.5	小结
449	
第17章	创建丰富的浏览器应用程序
450	
17.1	在标准浏览器中创建有活力的客户端应用程序
450	
17.1.1	asp.net ajax控件
451	
17.1.2	创建ajax页面
451	
17.1.3	asp.net ajax控件工具包开源库
454	
17.2	通过浏览器创造windows上独特和丰富的体验
459	
17.2.1	在单独的wpf和wpf xhap之间选择
459	
17.2.2	创建wpf浏览器应用程序
460	
17.2.3	安全考虑
461	
17.2.4	部署wpf浏览器应用程序
464	
17.3	跨平台提供交互式体验
467	
17.3.1	开始使用silverlight
467	
17.3.2	创建silverlight应用程序
468	

17.3.3	在网页中使用silverlight	475
17.4	小结	477
第18章	使用数据库	478
18.1	创建表和关系	478
18.1.1	新建sql server数据库	478
18.1.2	定义表	479
18.1.3	使用数据库关系图设计器	481
18.2	使用sql语句	483
18.2.1	编写查询	484
18.2.2	创建视图	486
18.2.3	开发存储过程	486
18.2.4	创建触发器	489
18.2.5	创建用户定义函数	490
18.3	使用数据库项目	490
18.3.1	创建数据库项目	490
18.3.2	自动生成脚本	491
18.3.3	执行脚本	492
18.4	在托管代码中创建数据库对象	492
18.4.1	创建sql server项目	492
18.4.2	用c#创建存储过程	493
18.5	将控件绑定到数据	495
18.5.1	数据绑定简介	495
18.5.2	自动生成windows forms控件绑定	496
18.5.3	编辑类型化数据集	500
18.5.4	手动绑定windows forms控件	

500	
18.5.5	web控件上的数据绑定
503	
18.6	对象关系映射
506	
18.6.1	linq概述
507	
18.6.2	使用o/r设计器映射
507	
18.6.3	linq代码
509	
18.7	小结
511	
第19章	面向服务的应用程序
512	
19.1	服务的基础知识
512	
19.2	asp.net web服务应用程序
514	
19.2.1	asp.net web服务项目模板
515	
19.2.2	创建asp.net web服务
517	
19.2.3	使用asp.net web服务
529	
19.2.4	管理web服务中的异常
533	
19.3	wcf服务应用程序
533	
19.3.1	wcf项目模板
534	
19.3.2	创建wcf服务
536	
19.3.3	配置wcf服务
540	
19.3.4	使用wcf服务
545	
19.3.5	安放和部署wcf服务
546	
19.4	小结
547	
第20章	为应用程序嵌入工作流
548	
20.1	wf
548	
20.1.1	工作流组件
549	
20.1.2	工作流项目组件
549	

- 20.1.3 workflow设计器
550
- 20.1.4 workflow项模板
552
- 20.1.5 workflow活动
554
- 20.2 创建顺序workflow
 - 20.2.1 设计workflow
555
 - 20.2.2 配置codeactivity
560
 - 20.2.3 配置ifelse活动和使用ruleconditioneditor
560
 - 20.2.4 配置invokewebservice活动
561
 - 20.2.5 配置handleexternalevent活动
562
 - 20.2.6 定义workflow宿主和客户
564
 - 20.2.7 运行workflow
570
- 20.3 创建状态机workflow
 - 20.3.1 设计状态机workflow
571
 - 20.3.2 初始化和转换状态
573
 - 20.3.3 定义客户与宿主
579
 - 20.3.4 运行状态机workflow
582
- 20.4 小结
583
- 第21章 开发office业务应用程序
584
 - 21.1 office扩展功能概述
585
 - 21.1.1 office功能
585
 - 21.1.2 visual studio office项目类型
587
 - 21.2 创建office外接程序
588
 - 21.2.1 自定义功能区
589
 - 21.2.2 自定义任务窗格
591
 - 21.2.3 创建outlook表单区域

592	
21.3	创建office文档扩展
594	
21.3.1	宿主控件
594	
21.3.2	创建操作窗格
596	
21.3.3	在数据缓存中存放数据
597	
21.3.4	实现自己的智能标签
598	
21.4	小结
601	
第六部分	vsts
第22章	团队协作和vsts
604	
22.1	软件开发项目概览
604	
22.1.1	msf agile
605	
22.1.2	msf for cmmi
606	
22.2	vsts简介
607	
22.2.1	vsts架构版
608	
22.2.2	vsts开发版
608	
22.2.3	vsts测试版
610	
22.2.4	vsts数据库版
611	
22.2.5	tfs
611	
22.3	小结
612	
第23章	管理和使用团队项目
614	
23.1	tfs剖析
614	
23.1.1	应用程序层
614	
23.1.2	数据层
616	
23.1.3	安全
617	
23.2	管理团队项目
619	
23.2.1	新建团队项目

619	
23.2.2	向项目团队中添加用户
621	
23.2.3	控制项目结构和迭代
624	
23.3	参与项目团队
625	
23.3.1	与tfs连接
625	
23.3.2	使用团队资源管理器
626	
23.3.3	使用项目门户
626	
23.3.4	使用microsoft office
627	
23.3.5	使用项目警告
628	
23.3.6	使用项目报表
629	
23.4	小结
630	
第24章	源代码控制
631	
24.1	team foundation source control基础
631	
24.1.1	基本架构
632	
24.1.2	安全权限和许可
632	
24.2	开始使用team foundation source control
633	
24.2.1	配置visual studio
634	
24.2.2	使用源代码控制资源管理器窗口
634	
24.2.3	管理工作区
635	
24.2.4	向源代码控制中添加文件
637	
24.3	在源代码控制中编辑文件
638	
24.3.1	从代码仓库中获取文件
638	
24.3.2	签入变更
638	
24.3.3	变更集的概念
642	
24.3.4	搁置代码
643	

24.3.5	合并更改	644
24.4	分支和合并	646
24.4.1	分支	647
24.4.2	合并	647
24.5	小结	648
第25章	工作项跟踪	649
25.1	工作项	650
25.1.1	工作项的角色和sdic	650
25.1.2	为项目选择工作项集	650
25.1.3	工作项的共性	653
25.2	使用team explorer管理工作项	659
25.2.1	新建工作项	659
25.2.2	查找和过滤工作项	661
25.3	团队角色透视	664
25.3.1	项目的视角	664
25.3.2	项目经理	664
25.3.3	业务分析师	670
25.3.4	开发人员	671
25.3.5	测试人员	674
25.4	自定义工作项	675
25.4.1	将工作项植入流程	675
25.4.2	自定义现有工作项	679
25.5	小结	680
第26章	开发版	681
26.1	可视化代码开发	

681	
26.1.1	class designer
682	
26.1.2	添加项
683	
26.1.3	定义类之间的关系
684	
26.1.4	定义方法、属性、字段和事件
686	
26.2	开发人员的单元测试
686	
26.2.1	单元测试示例
687	
26.2.2	编写有效的单元测试
687	
26.2.3	使用单元测试类和方法
688	
26.2.4	创建单元测试
689	
26.2.5	运行单元测试
690	
26.2.6	代码覆盖率分析
692	
26.3	性能分析
694	
26.3.1	创建性能会话
694	
26.3.2	配置会话
696	
26.3.3	会话目标
699	
26.3.4	报表
700	
26.3.5	性能报告
700	
26.4	代码分析
706	
26.4.1	配置强制规则
707	
26.4.2	把规则违例视为编码错误
707	
26.4.3	禁止规则
708	
26.4.4	分析
710	
26.4.5	查看结果
710	
26.4.6	更新说明性指导
711	

26.5	代码度量	711
26.5.1	测量法	711
26.5.2	运行代码度量	712
26.5.3	处理结果	712
26.6	小结	714
第27章	架构版	715
27.1	team architect制品	715
27.1.1	项目模板	716
27.1.2	项模板	716
27.2	设计应用程序	717
27.3	定义系统	723
27.4	定义基础结构	725
27.5	部署应用程序	731
27.5.1	部署关系图	731
27.5.2	验证部署	732
27.5.3	部署报表	733
27.6	实现应用程序	733
27.6.1	设置实现属性	733
27.6.2	生成项目	734
27.7	小结	735
第28章	测试版	736
28.1	创建、配置和管理测试	736
28.1.1	测试项目	737
28.1.2	测试项	738
28.1.3	测试管理	

739	
28.1.4	测试配置
741	
28.2	web测试
741	
28.2.1	记录web测试
741	
28.2.2	管理web测试请求
743	
28.2.3	运行web测试和查看结果
743	
28.2.4	为web测试设置数据
744	
28.2.5	从web测试中提取值
749	
28.2.6	请求验证规则
750	
28.3	负载测试
751	
28.3.1	创建负载测试
751	
28.3.2	复审和编辑负载测试
756	
28.3.3	运行负载测试和检查结果
756	
28.4	手动测试
758	
28.4.1	创建手动测试
758	
28.4.2	执行手动测试
758	
28.5	一般测试
759	
28.6	顺序测试
759	
28.7	小结
760	
第29章	数据库版
761	
29.1	数据库项目系统
761	
29.1.1	创建数据库项目
762	
29.1.2	schema view
767	
29.2	比较架构
768	
29.2.1	查看对象定义
769	

29.2.2	架构更新脚本	770
29.2.3	比较选项	771
29.3	比较数据	771
29.3.1	浏览记录级别的详细信息	772
29.3.2	预览和执行更新脚本	773
29.4	重命名重构	774
29.4.1	重命名选项	774
29.4.2	预览架构更改	774
29.5	单元测试	775
29.5.1	创建单元测试	775
29.5.2	数据库单元测试设计器	777
29.6	生成数据	779
29.6.1	创建数据生成计划	779
29.6.2	预览生成的数据	781
29.6.3	生成数据	783
29.7	生成和部署	784
29.8	小结	785
第30章	team foundation build	786
30.1	team foundation build简介	786
30.2	新建生成	789
30.2.1	为生成命名	789
30.2.2	选择要生成的项目文件	790
30.2.3	定义生成项目文件	790
30.2.4	定义生成滞留策略	792
30.2.5	定义生成代理配置	

792	
30.2.6	生成计划或设置生成触发器
793	
30.3	编辑生成定义
794	
30.3.1	tfsbuild.proj项目文件
795	
30.3.2	msbuild的作用
795	
30.4	启动生成
795	
30.5	监视和分析生成
796	
30.5.1	team build explorer简介
796	
30.5.2	生成报表
797	
30.6	小结 798

章节摘录

当初步了解.NET的语言无关性后，开发者会提出许多问题。其中最普遍的问题可能就是：“如果所有的.NET语言都会编译成‘托管代码’，为什么我们还需要多种编译器呢？”这个问题有多个答案。首先，程序员在选择编程语言时有各自不同的喜好（包括我自己）。一些人喜欢充满分号和圆括号而且关键字相当少的语言；另一些人喜欢更具有“可读性”语法标记的语言（如EIVB）；还有一些人在开始转向.NET平台时还希望可以使用他们已掌握的技能（通过COBOL.NET）。现在，平心而论，如果微软推出一门派生自BASIC语言系列的“官方”.NET语言，你认为所有的程序员会喜欢这样的选择吗？或者，如果这个唯一的“官方”.NET语言是基于Fortran语法的，那么可以想象所有人都会对.NET置之不理。因为.NET运行库并不在意一段托管代码是由哪种语言生成的，所以.NET程序员可以继续使用他们熟悉的语法，且与组员、部门甚至其他公司共享编译的程序集（不管他们用的是哪种.NET语言）。将各种.NET语言集成为一个统一软件方案的另一个好处，就是能够取长补短。所有的编程语言都有各自的优点和缺点。例如，一些编程语言对高级的数学处理有相当完美的内在支持能力。另一些则精于支持财务计算、逻辑计算和与大型机交互等。当你学习到某种编程语言的优点并将其融合于.NET平台时，大家就都能受益。当然，实际上我们大部分时间还是在用自己习惯的.NET语言来编写程序。但是，一旦学会了一种.NET语言的语法，就很容易掌握其他的了。这是非常有益的，对软件技术顾问而言尤其如此。如果你熟悉C#，在为只使用Visual Basic.NET的客户做咨询时，你仍然能够使用.NET Framework的功能，并且可以毫不费力地掌握代码的整体结构。够棒的吧。

1.5.NET程序集概览不管选择了哪种.NET语言编程，需要明白的是，尽管.NET二进制文件COM服务器和非托管Win32二进制文件（*.dll或*.exe）具有相同的文件扩展名，但它们的内部却是完全不同的。例如，*.dll的.NET二进制文件不会导出与COM运行库进行通信的方法（因为.NET不是COM）。此外，.NET二进制文件不使用COM类型库文件描述而且不用在系统注册表中注册。也许更重要的是，.NET二进制文件不包含特定于平台的指令，它包含的是平台无关的IL（中间语言）和类型元数据。图1.3清楚显示了这个流程。

插图：

《C#与.NET 3.5高级程序设计》

编辑推荐

《C#与.NET 3.5高级程序设计》(第4版)由微软C# MVP Andrew Troelsen编写,历经多次修订,适合各层次.NET开发人员阅读。Amazon超级畅销书,全面涵盖.c#3.O,用IL深入揭示各语言特性,深度剖析.NET 3.5平台。让你知其然,更知其所以然,国内多位微软MVP联手翻译。C#语言作为.NET平台上的第一语言,已经成为目前功能最强大的通用语言之一。《C#与.NET 3.5高级程序设计(第4版)》是被誉为“C#圣经”的经典巨著,因语言生动流畅、剖析深入、涵盖全面而广受推崇。畅销不衰。曾经获得Referenceware编程图书大奖。并入选Jolt大奖提名。书中探讨了C#语言和.NET平台的各种特性。包括重载运算符、指针、泛型等高级功能和Clt、远程处理、Windows Forms、ASP.NET、ADO.NET等技术。不少概念都通过IL代码透视其背后的本质,使你知其然。更知其所以然。新版更透彻阐述了C#3.0新功能(包括自动属性、扩展方法、匿名类型等)和.NET 3.5的最新特性(包括LINQ、WPF、WCF和WF等相关技术)。附录中包括了COM与.NET的互操作和Mono发等主题。与同类图书不同。全书由世界级C#专家Andrew Troelsen以一人之力完成,因此写作思路和布局谋篇都独具匠心。中文版由国内多位微软MVP联手译出。强大的译者阵容有力地保证了这部权威原著原汁原味的重现。

精彩短评

- 1、内容与前言的溢美之词很不匹配，顶多当工具书用，但形式上又不如Core Java，在微软大旗下所谓的那些大师们，跟计算机科学领域的大师根本没法比
- 2、27号(周日)订货，31号早上到货，原本期待30号能收到的，虽然比期待晚了一天但还是很欣喜。书的外观完好无损，内页尚未发现破坏，一看上去书很新。粗略翻看了一下，内容多、全、深但不细，或者是我功力不够还不到能参悟的级别，将静心潜读。
- 3、感觉不是很好！毕竟是高级编程，不大适合初学者。网络编程的没有。
- 4、非常棒的一本书
- 5、无论是原版英文在内容方面的广度和深度，以及翻译文字的质量都是我所读的难遇的好书
- 6、言过其实，凑合当参考书
- 7、卓越的书不仅很正版，书的内容更加丰富，受益匪浅！
- 8、工具书类型的入门书籍
- 9、作者对ASP.net3.5非常内行，不仅介绍了功能，还介绍了微软的想法，原来微软的东西是这样用的！！如果你想成为微软工具开发的正规军，那这是一本了不起的书。
- 10、全面 简单 实用
- 11、内容很多，书很厚重
- 12、.net搞3年了，看了后感受很多，特别细节上，对于概念的理解，举的例子都很易懂，可以让人跟容易理解，至少比当初在学校时好多了书里写“源代码在XX目录下”是不是有光盘？我怎么没有收到？
- 13、书名叫“高级程序设计”，非常名副其实，太高级了
- 14、如果你在C#言语方面只想读一本书，那就选这本吧。最新[2014年]是第5版。
- 15、适合新手的C#入门与参考书，带有源码。但价格稍贵。
- 16、买了这本书，居然在预计送到日期前到货我跟踪下自己的订单，居然发现收到书时其还在配送途中，我怀疑快递公司是不是给我的是正版书啊！封面有点皱巴巴的。
- 17、哈灵~！！
- 18、很贵，但是很值得。。
- 19、用了张十元代金券，竟然给我一本旧书。书皮褶皱，侧面暗黄。哎~~~~不厚道。。。。。
- 20、前部分不错，后面的WCF、WF虽然提到了，但是和没提一样，还不是推荐你再去买一本书
- 21、名副其实的C#圣经
- 22、经典教材，不错。。
- 23、还是学到了不少东西
- 24、讲的东西太全了，如果是入门的话看的建议看核心技术。
- 25、老公买的，千挑万选选中的哦，老公觉得不错
- 26、此书购于10-08-20，花了6个月时间才读完~
- 27、相比C#入门经典，这本书讲解的相对深入。另外需要克服一个误解，跟着书中的示例去做可以帮助你初步理解所述概念，但是只有在做自己的项目时候，才能更主动去思索其所以然，挖掘其产生的深刻意义。
- 28、还没看，但发货很快，第二天就拿到手了，很厚，有的看了，只是没光盘
- 29、和原版对照着看，可以发现翻译的错误百出，有些地方的意思居然和原版相反，简直是误人子弟，由多位译者翻译的书，大都是滥竽充数之作，质量难以保证。
- 30、没读完
- 31、给LG买的。他说还可以。
- 32、从C#语言层面到.NET框架层面介绍了两者的特性和使用，对于新版本的语言特性和框架特性也有介绍，不错
- 33、和Wrox红皮书《C#高级编程》同一级别的书，但是个人认为更胜一筹。
- 34、讲的比较深入，吧编译的深层运行机制细细道来，不错。。
- 35、如果需要学习c#,那么就请读这本书吧，它让里从.net底层到高层全面了解c#这本书我最早是看英文电子版的，中文版出来就马上买了一本

精彩书评

- 1、很不错的一本书，对初学者来很有意义，虽说部分内容有点不好理解，但是不理解的可以先跳过去啦，慢慢看，收获一定不会少的~
- 2、这本书很好啊，通读一遍对整个.net平台会有一个整体的了解。相当不错。目前在看中文版的，等看完再看看英文的。
- 3、对于稍微有点.NET基础的读者，通读全书基本不会发现有什么难度，没有任何难理解的地方。对于.NET介绍的很全面，但是每一点都说的比较肤浅，刚刚说到稍微深入的地方时戛然而止，读起来总是感觉差一口气，意犹未尽的感觉。适合初级学习，对.NET有一个总体把握和理解。也可以当作字典收藏，时常拿出来翻翻，对于解决问题说不定会有新的灵感。但是只讲应用不讲原理，读起来实在有点不爽！
- 4、很不好意思的说，我是一名大一的学生，刚看完一本WPF的书，是C#里的纯正菜鸟级别，所以有什么失误之处还请各位大侠原谅。我觉得这本书是很好很好的，很全面，但似乎过于全面了，各种语言一起上，搞得我等没什么编程语言的小辈无言以对了。但我还是由衷的感谢作者，给了我很多帮助！

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:www.tushu111.com