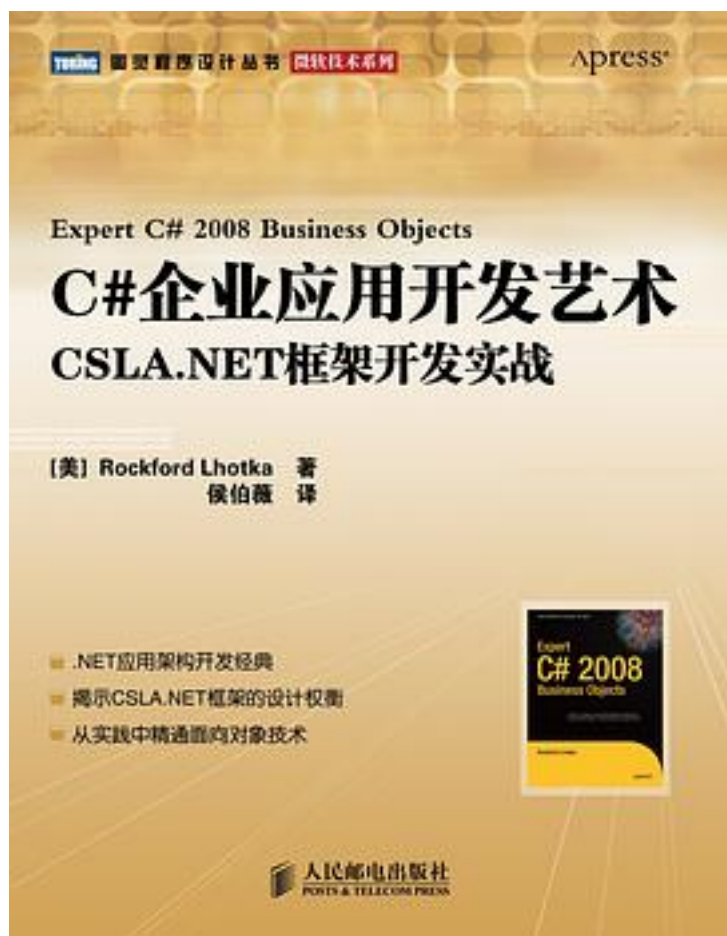


# C#企业应用开发艺术



[C#企业应用开发艺术\\_下载链接1](#)

著者:Rockford Lhotka

出版者:人民邮电出版社

出版时间:2010-4

装帧:平装

isbn:9787115222206

CSLA.NET是目前.NET平台上最广泛使用的开发框架之一。使用这一框架，开发人员不必过于关心底层细节，而可以集中精力考虑业务逻辑和应用设计。本书展现了设计和创建CSLA.NET框架的整个思考过程。

围绕这一框架，作者讲述了怎样用面向对象的思想来搭建.NET应用程序的架构，设计和开发.NET应用，重点介绍如何创建专注于业务的对象，使其适应于各种分布式环境；如何使用它们来搭建基于WPF、Web窗体、WCF、Windows窗体、WF的应用程序。书中还以一个包含多个接口的实例应用，教会读者怎样用好这一框架。无论你在开发中是否采用这一框架，都能从本书受益匪浅。

作者介绍:

Rockford Lhotka 微软软件传奇人物、微软Regional Director、MVP、INETA发言人，经常在许多国际性会议和用户组大会上发表精彩演讲，是MSDN在线的专栏作家。他还是微软金牌认证合作伙伴Magenic技术公司的传道者。

目录: 第1章 分布式架构 1

1.1 逻辑和物理架构 1

1.1.1 N层和SOA 3

1.1.2 复杂性 3

1.1.3 逻辑模型和物理模型之间的关系 4

1.1.4 5层逻辑架构 8

1.1.5 应用逻辑框架 12

1.1.6 展望未来 15

1.2 管理业务逻辑 15

1.2.1 可能的业务逻辑位置 16

1.2.2 业务对象 18

1.2.3 移动对象 21

1.3 架构和框架 27

1.4 小结 28

第2章 框架设计 29

2.1 基本设计目标 30

2.1.1 验证规则和业务规则 31

2.1.2 跟踪对象是否改变 32

2.1.3 集成授权 32

2.1.4 子对象的强类型集合 33

2.1.5 多级撤销能力 34

2.1.6 用户界面开发者的简单抽象模型 35

2.1.7 支持数据绑定 40

2.1.8 对象持久性和对象—关系映射 43

2.1.9 自定义身份验证 48

2.2 设计框架 49

2.2.1 创建业务对象 49

2.2.2 多级撤销功能 56

2.2.3 数据绑定支持 58

2.2.4 业务和验证规则 60

2.2.5 数据门户 62

2.2.6 自定义身份验证 74

2.2.7 整合授权 75

2.2.8 辅助类型和类 76

2.3 命名空间组织 79

2.4 小结 80

第3章 面向对象应用程序设计 82

3.1 责任驱动设计 83

3.1.1 用例或基于故事的分析 83

3.1.2 带有责任的对象	83
3.1.3 为用例存在的对象	83
3.1.4 行为的标准化	84
3.2 应用程序需求	84
3.3 对象设计	88
3.3.1 最初的设计	88
3.3.2 修订设计	89
3.3.3 自定义身份验证	99
3.4 使用CSLA.NET	99
3.5 数据库设计	102
3.5.1 创建数据库	103
3.5.2 PTracker数据库	104
3.5.3 Security数据库	112
3.6 小结	113
第4章 CSLA.NET对象构造型	114
4.1 基本术语和对象图的结构	114
4.2 业务对象生命周期	116
4.2.1 对象的创建	117
4.2.2 取得对象	121
4.2.3 更新可编辑的对象	123
4.2.4 销毁和终结对象	128
4.3 业务类结构	130
4.3.1 Serializable或DataContract特性	130
4.3.2 通用区域	132
4.3.3 非公有的默认构造函数	136
4.4 小结	136
第5章 CSLA.NET对象模板	137
5.1 业务类的结构	137
5.1.1 可编辑的根业务对象	137
5.1.2 可编辑的子业务对象	143
5.1.3 可切换的对象	147
5.1.4 可编辑的根集合	150
5.1.5 可编辑的子集合	152
5.1.6 只读的业务对象	153
5.1.7 只读子对象	155
5.1.8 只读集合	156
5.1.9 只读子集合	157
5.1.10 命令对象	158
5.1.11 名称/值列表对象	159
5.1.12 动态可编辑集合	161
5.1.13 动态可编辑根对象	162
5.1.14 条件对象	163
5.2 小结	165
第6章 业务框架实现	166
6.1 CSLA.NET项目结构	167
6.1.1 项目目录结构	167
6.1.2 项目设定	167
6.1.3 项目签名	168
6.1.4 支持本地化	169
6.2 Csla命名空间	170
6.2.1 ApplicationContext	171
6.2.2 BusinessBase	175
6.2.3 BusinessListBase	177
6.2.4 CommandBase	179

- 6.2.5 CriteriaBase 179
- 6.2.6 DataPortal 180
- 6.2.7 EditableRootListBase 180
- 6.2.8 NameValueListBase 180
- 6.2.9 PropertyInfo 181
- 6.2.10 ReadOnlyBase 182
- 6.2.11 ReadOnlyListBase 183
- 6.2.12 SingleCriteria 184
- 6.2.13 SmartDate 184
- 6.2.14 Utilities 185
- 6.3 Csla.Core命名空间 185
  - 6.3.1 BusinessBase 186
  - 6.3.2 ExtendedBindingList 186
  - 6.3.3 IBusinessObject接口 187
  - 6.3.4 ICommandObject接口 188
  - 6.3.5 IEditableBusinessObject接口 188
  - 6.3.6 IEditableCollection接口 188
  - 6.3.7 IReadOnlyObject接口 188
  - 6.3.8 IReadOnlyCollection接口 188
  - 6.3.9 ISavable接口 189
  - 6.3.10 ISmartField接口 189
  - 6.3.11 ISupportUndo接口 190
  - 6.3.12 ITrackStatus接口 190
  - 6.3.13 IUndoableObject接口 190
  - 6.3.14 ObjectCloner类 191
  - 6.3.15 ReadOnlyBindingList 192
- 6.4 小结 193
- 第7章 属性声明 194
  - 7.1 声明属性 194
    - 7.1.1 属性声明的选项 196
    - 7.1.2 RegisterProperty和继承 200
  - 7.2 PropertyInfoManager 202
  - 7.3 字段管理器 204
    - 7.3.1 FieldManager属性 204
    - 7.3.2 FieldDataManager类 205
  - 7.4 小结 211
- 第8章 对象状态管理 212
  - 8.1 对象状态属性 212
    - 8.1.1 ITrackStatus接口 212
    - 8.1.2 IsNew 213
    - 8.1.3 IsSelfDirty 214
    - 8.1.4 IsDirty 217
    - 8.1.5 IsSelfValid 217
    - 8.1.6 IsValid 218
    - 8.1.7 IsSavable 218
    - 8.1.8 IsDeleted 219
  - 8.2 小结 220
- 第9章 父子关系 221
  - 9.1 可编辑的父对象 221
    - 9.1.1 父子对象之间的交互 222
    - 9.1.2 IParent接口 226
    - 9.1.3 声明Child属性 226
  - 9.2 可编辑的父集合 229

- 9.3 小结 232
- 第10章 数据绑定 233
  - 10.1 Windows窗体 233
    - 10.1.1 对象数据绑定 233
    - 10.1.2 集合数据绑定 240
    - 10.1.3 控件和辅助对象 240
    - 10.1.4 与多个根对象协同工作 254
  - 10.2 WPF 255
    - 10.2.1 对象数据绑定 255
    - 10.2.2 集合数据绑定 257
    - 10.2.3 控件和帮助对象 258
  - 10.3 Web窗体 269
  - 10.4 小结 271
- 第11章 业务和验证规则 272
  - 11.1 规则类型 272
  - 11.2 Csla.Validation命名空间 273
    - 11.2.1 RuleHandler委托 273
    - 11.2.2 RuleArgs类 274
    - 11.2.3 DecoratedRuleArgs类 274
    - 11.2.4 RuleMethod类 275
    - 11.2.5 RuleDescription类 277
    - 11.2.6 ValidationRules类 277
    - 11.2.7 BrokenRule类 283
    - 11.2.8 BrokenRulesCollection类 284
    - 11.2.9 ValidationException 286
  - 11.3 通用验证规则 286
  - 11.4 小结 290
- 第12章 身份验证和授权 291
  - 12.1 身份验证 291
    - 12.1.1 Csla.ApplicationContext.  
User属性 292
    - 12.1.2 Windows身份验证 292
    - 12.1.3 自定义身份验证 293
  - 12.2 授权 297
    - 12.2.1 类型级别授权 297
    - 12.2.2 属性和方法级别的授权 300
  - 12.3 小结 304
- 第13章 多级撤销 305
  - 13.1 使用撤销 306
  - 13.2 实现撤销 306
    - 13.2.1 ISupportUndo接口 307
    - 13.2.2 NotUndoableAttribute类 308
    - 13.2.3 UndoableBase类 308
    - 13.2.4 BusinessBase类 317
    - 13.2.5 BusinessListBase类 319
  - 13.3 小结 329
- 第14章 LINQ to CSLA 330
  - 14.1 使用LINQ降低代码量 330
  - 14.2 LINQ to CSLA.NET概览 331
    - 14.2.1 将来自于LINQ to Objects的结果进行绑定 331
    - 14.2.2 索引的LINQ查询 331
  - 14.3 LINQ和投影 331
    - 14.3.1 标识投影和  
LinqBindingList<T> 332

- 14.3.2 理解LinqBindingList 332
- 14.4 使用CSLA.NET进行索引检索的概览 333
  - 14.4.1 序列化和索引 334
  - 14.4.2 索引模式 334
- 14.5 CSLA.NET的IQueryable实现 335
  - 14.5.1 理解表达式树 335
  - 14.5.2 深入探索IQueryProvider 336
  - 14.5.3 LinqBindingList 338
- 14.6 被索引的LINQ和CSLA.NET 338
  - 14.6.1 管理索引集 338
  - 14.6.2 表达式求值 339
  - 14.6.3 索引对象模型 340
- 14.7 小结 345
- 第15章 持久性和数据门户 346
  - 15.1 数据门户设计 346
    - 15.1.1 业务逻辑和数据访问的分离 347
    - 15.1.2 根对象和子对象的统一编码模型 347
    - 15.1.3 通道适配器和消息路由器模式 348
    - 15.1.4 分布式事务支持 352
    - 15.1.5 上下文和位置透明性 353
    - 15.1.6 授权服务调用 356
    - 15.1.7 异步行为 357
    - 15.1.8 对象工厂 357
  - 15.2 基类的支持 358
    - 15.2.1 工厂方法和条件 359
    - 15.2.2 Save方法 361
    - 15.2.3 使用字段管理器更新子对象 364
    - 15.2.4 更新可编辑的集合 365
  - 15.3 反射和动态方法调用 366
    - 15.3.1 MethodCaller类 366
    - 15.3.2 LateBoundObject类 367
  - 15.4 通道适配器 368
    - 15.4.1 RunLocal特性 368
    - 15.4.2 DataPortal类 369
    - 15.4.3 DataPortal<T>类 377
    - 15.4.4 IDataPortalServer接口 379
    - 15.4.5 IDataPortalProxy接口 380
    - 15.4.6 LocalProxy类 380
    - 15.4.7 WcfProxy类 381
    - 15.4.8 WcfPortal类 382
  - 15.5 分布式事务支持 383
    - 15.5.1 Transactional特性 383
    - 15.5.2 Csla.Server.DataPortal对象 384
    - 15.5.3 ServicedDataPortal类 387
    - 15.5.4 TransactionalDataPortal类 389
  - 15.6 消息路由器 389
    - 15.6.1 DataPortalSelector类 390
    - 15.6.2 SimpleDataPortal类 391
    - 15.6.3 FactoryDataPortal类 396
    - 15.6.4 FactoryLoader属性 396
    - 15.6.5 ChildDataPortal类 399
  - 15.7 上下文和位置透明性 400
    - 15.7.1 DataPortalContext类 400

- 15.7.2 DataPortalResult类 403
- 15.7.3 Csla.Server.DataPortal-Exception 403
- 15.8 小结 404
- 第16章 其他框架特性 405
- 16.1 使用SmartDate处理日期 406
  - 16.1.1 初始化结构体 408
  - 16.1.2 支持空日期 408
  - 16.1.3 转换函数 409
  - 16.1.4 文本函数 410
  - 16.1.5 日期函数 411
  - 16.1.6 数据库格式 411
- 16.2 数据访问 412
  - 16.2.1 管理数据库连接和上下文 412
  - 16.2.2 SafeDataReader 417
  - 16.2.3 DataMapper 420
- 16.3 Windows工作流基础 425
  - 16.3.1 从对象中启动工作流 426
  - 16.3.2 WorkflowManager类 426
- 16.4 小结 431
- 第17章 对业务对象的实现 432
- 17.1 ProjectTracker对象 432
- 17.2 设置项目 433
- 17.3 业务类的实现 434
  - 17.3.1 Project 434
  - 17.3.2 ProjectResources 443
  - 17.3.3 ProjectResource 445
  - 17.3.4 Assignment 447
  - 17.3.5 RoleList 448
  - 17.3.6 Resource和相关的对象 449
  - 17.3.7 ProjectList和ResourceList 450
  - 17.3.8 Roles 451
  - 17.3.9 Role 453
  - 17.3.10 实现Exists方法 455
- 17.4 自定义身份验证 455
  - 17.4.1 PTPrincipal 455
  - 17.4.2 PTIdentity 457
- 17.5 小结 458
- 第18章 数据访问的示例 459
- 18.1 数据访问层设计 459
  - 18.1.1 数据访问模型 459
  - 18.1.2 平衡设计问题 465
- 18.2 数据访问对象 467
  - 18.2.1 使用LINQ to SQL 467
  - 18.2.2 ProjectTracker.DalLinq项目 468
- 18.3 业务类的实现 469
  - 18.3.1 Project 470
  - 18.3.2 ProjectResources 478
  - 18.3.3 ProjectResource 479
  - 18.3.4 RoleList 482
  - 18.3.5 ProjectList和ResourceList 484
  - 18.3.6 Roles 485
  - 18.3.7 实现Exists方法 487
- 18.4 小结 488

- 第19章 WPF用户界面 490
  - 19.1 WPF中的自定义身份验证 490
  - 19.2 界面设计 491
    - 19.2.1 用户控件框架 492
    - 19.2.2 值转换器 496
    - 19.2.3 应用程序配置 498
    - 19.2.4 PTWpf项目设定 501
    - 19.2.5 MainForm窗口 501
    - 19.2.6 登录窗口 504
    - 19.2.7 RolesEdit窗体 505
    - 19.2.8 ResourceList窗体 512
    - 19.2.9 ProjectList窗体 513
    - 19.2.10 ProjectEdit窗体 514
  - 19.3 小结 517
- 第20章 Web窗体用户界面 518
  - 20.1 Web开发和对象 518
  - 20.2 状态管理 520
    - 20.2.1 将状态存储在Web服务器上 521
    - 20.2.2 将状态传送到客户端，或从客户端传回 523
    - 20.2.3 状态在文件或者数据库中 523
  - 20.3 界面设计 524
    - 20.3.1 应用程序配置 526
    - 20.3.2 PTWeb站点设置 529
    - 20.3.3 母版页 530
    - 20.3.4 登录页面 534
  - 20.4 业务功能 540
    - 20.4.1 RolesEdit窗体 540
    - 20.4.2 ProjectList窗体 549
    - 20.4.3 ProjectEdit窗体 552
  - 20.5 小结 562
- 第21章 WCF服务界面 564
  - 21.1 客户端/服务器还是SOA 564
  - 21.2 WCF服务概览 565
    - 21.2.1 WCF服务的元素 566
    - 21.2.2 自定义身份验证 572
  - 21.3 设计WCF服务界面 585
    - 21.3.1 服务设计 586
    - 21.3.2 应用程序配置 586
    - 21.3.3 PTWcfService站点设置 588
    - 21.3.4 服务契约和实现 589
  - 21.4 Web服务消费应用程序实现 594
    - 21.4.1 生成WCF信息页面 594
    - 21.4.2 简单的智能客户端 595
  - 21.5 小结 599
- • • • • (收起)

[C#企业应用开发艺术\\_下载链接1](#)

标签



架构

C

#程序设计

asp.net

N-TIER框架设计

计算机科学

计算机技术

计算机

## 评论

侯伯微翻译的太晦涩，中国人看不懂的中文。

-----  
一下子就读完了，这种书以后还是图书馆借比较划算。

-----  
看的第一本架构书

-----  
不过最好是看英文版本的，中文翻译。。。 （高级汉语8级水准才能看懂）

-----  
刚刚买了,但是还没时间读,打算把大话数据结构读完就接着读它

-----  
[C#企业应用开发艺术\\_下载链接1](#)

## 书评

CSLA.NET是目前.NET平台上最广泛使用的开发框架之一。使用这一框架，开发人员不必过于关心底层细节，而可以集中精力考虑业务逻辑和应用设计。本书展现了设计和创建CSLA.NET框架的整个思考过程。

围绕这一框架，作者讲述了怎样用面向对象的思想来搭建.NET应用程序的架构，设计和...  
...

-----  
翻译的语句读起来很晦涩 例如将interface翻译为界面，个人觉得不太妥当 separate application我更愿意理解为一种“剥离的应用程序”而不是“分离的应用程序”

另外例如：

该层包含以下逻辑：决定用户看到什么，对路径的导航（这是个什么虾米），以及解释用户输入。我愿意翻译...

-----  
这本书讲的主要是软件的架构,对于架构方面,本人还有很大的成长空间,所以刚刚买了一本,但是还没时间读,打算把大话数据结构读完就接着读它.

粗略看了下架构,感觉很不错,它让体会到了架构的美妙,而且它是适合有一定技术基础的人,所以对我有点挑战性

-----  
[C#企业应用开发艺术\\_下载链接1](#)