第一图书网, tushu007.com

<<LINQ实战>>

图书基本信息

书名：<<LINQ实战>>

13位ISBN编号：9787115208514

10位ISBN编号：7115208514

出版时间：2009-07

出版时间：人民邮电出版社

作者：Fabrice Marguerie,Steve Eichert,Jim Wooley

页数：416

译者：陈黎夫

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：http://www.tushu007.com

前言

提笔为这本书写序并不容易，不是因为回首望去，LINO的开发历程漫长而又千辛万苦；也不是因为我此时此刻热泪盈眶，心潮澎湃；也并非因为我必须字斟句酌地来总结这个倾注了我全部心血的产品。
之所以说不容易，是因为这定会成为一本备受推崇的书，让我很难揣摩出一个恰如其分的点睛之笔去形容它。
对我而言，LINQ项目早在发布任何官方消息数年之前就已经悄然开始了，那时我正在参与一个新的托管ADO的计划和架构设计。
当时，几个非常聪明的开发者非常大胆地建议，让我们将那些传统的数据访问API统统扔掉，而是基于对象和元数据进行设计，因为元数据和对象是最新的运行时（即Java运行时）的基石。
不过这个建议并没有被采纳——传统的想法占据了上风，那时我也坚定地站在了传统的一方。
但由此我理解了数据应该处于任何编程系统的核心，而不仅仅是其附庸。
有理由认为，在基于对象的系统中，应该让数据也作为对象。
不过，要实现这个理想，却注定要经历九九八十一难。
作为一名工程师，我时刻位于微软公司各种新技术的风口浪尖上，我们要设计新的API、创造新的语言特性，带动技术向前发展。
虽然很多想法从未有机会变成最终的产品，但每次尝试却都是迈向正确方向时不可或缺的一步。
LINQ正是这些努力、斗争以及教训的最终结晶。
它凝聚着团队成员敏锐的洞察力和对计算机科学理论及技术的深入理解，而且若是缺少了AndersHejlsberg的睿智与专注细节，LINQ也不可能成为现实。
当然，其中还有你们大家的努力。
开发者社区在论坛和博客上的讨论对LINO的成形产生了极为深远的影响。
能够在开发的过程中就得到如此迅捷的反馈，不亚于在漆黑一片的房间中点起了一盏明灯。
更让人高兴的是看到星星之火得以燎原，许多开发者逐步成长为专家、传道者，开始做讲座、写文章，大家相互交流，彼此启发。

内容概要

作为.NET上连接编程语言和数据库、内存对象、XML等各种类型数据之间的桥梁，LINQ引入了一种处理数据的全新理念，将查询无缝集成至开发语言之上。
《LINQ实战》第一部分介绍了LINQ技术及C#和VB为此提供的新语言特性，第二、三、四部分详细介绍了LINQ的三大主要功能LINQ to Objects、LINQ to SQL、LINQ to XML，第五部分研究了LINQ的可扩展性、在应用程序中使用各种LINQ功能的方法以及LINQ to DataSet。
《LINQ实战》适合于使用C#和VB.NET的软件开发者。

第一图书网, tushu007.com

作者简介

作者：(法国) 马格里 (Marguerie.F.) (美国) 艾歇特 (Eichert.S.) (美国) 伍利 (Wooley.J.) 译者：陈黎夫Fabrice Marguerie，微软C#MVP，资深NET软件架构师，LINQ社区专家。
Steve Eichert，资深程序员，擅长.NET与Ruby，有十多年编程经验。
Jim Wooley，微软MVP，LINQ社区专家。

书籍目录

第一部分 从这里开始第1章 LINQ简介1.1 LINQ是什么1.1.1 概览1.1.2 LINQ的功能1.1.3 作为语言扩展的LINQ1.2 为什么需要LINQ1.2.1 常见问题1.2.2 解决理念上的失调1.2.3 LINQ给出解决方案1.3 LINQ的设计目标及起源1.3.1 LINQ项目的设计目标1.3.2 一点历史1.4 LINQtoObjects初步：查询内存中的集合1.4.1 先决条件1.4.2 HelloLINQtoObjects1.5 LINQtoXML初步：查询XML文档1.5.1 为什么需要LINQtoXML1.5.2 HelloLINQtoXML1.6 LINQtoSQL初步：查询关系型数据库1.6.1 LINQtoSQL功能概览1.6.2 HelloLINQtoSQL1.6.3 进一步了解LINQtoSQL1.7 小结第2章 C#和VB语言的增强特性2.1 新语言的增强特性一瞥2.1.1 列出当前正在运行的所有进程2.1.2 将返回结果封装成类2.2 隐式类型局部变量2.2.1 语法2.2.2 使用隐式类型局部变量改进实例程序2.3 对象和集合的初始化器2.3.1 引入对象初始化器的动机2.3.2 集合初始化器2.3.3 用对象初始化器改进实例程序2.4 Lambda表达式2.4.1 委托回顾2.4.2 匿名方法2.4.3 Lambda表达式介绍2.5 扩展方法2.5.1 创建扩展方法的示例2.5.2 更多使用LINQ标准查询操作符的例子2.5.3 将扩展方法引入到实例程序中2.5.4 注意事项2.6 匿名类型2.6.1 使用匿名类型组合数据2.6.2 没有名字的类型也是类型2.6.3 使用匿名类型改进实例程序2.6.4 匿名类型的限制2.7 小结第3章 LINQ构建块3.1 LINQ对.NET的扩展3.1.1 语言扩展回顾3.1.2 构成LINQ基础的核心元素3.2 序列3.2.1 IEnumerable接口3.2.2 迭代器回顾3.2.3 延迟查询执行3.3 查询操作符3.3.1 查询操作符是什么3.3.2 标准查询操作符3.4 查询表达式3.4.1 查询表达式是什么3.4.2 编写查询表达式3.4.3 标准查询操作符与查询表达式的关系3.4.4 限制3.5 表达式树3.5.1 Lambda表达式回顾3.5.2 表达式树是什么3.5.3 IQueryable，另一种实现延迟查询执行的方法3.6 LINQ的程序集以及命名空间3.7 小结第二部分 查询内存中的对象第4章 熟悉LINQtoObjects4.1 实例程序简介4.1.1 程序目标4.1.2 程序功能4.1.3 业务逻辑实体4.1.4 数据库架构4.1.5 示例数据4.2 用LINQ操作内存中的集合4.2.1 能够查询什么4.2.2 支持的操作4.3 在ASP.NET和WindowsForms中使用LINQ4.3.1 Web应用程序中的数据绑定4.3.2 WindowsForms应用程序中的数据绑定4.4 常用的标准查询操作符4.4.1 约束操作符Where4.4.2 投影操作符4.4.3 Distinct操作符4.4.4 转换操作符4.4.5 聚合操作符4.5 用不同的方式显示内存中的集合4.5.1 排序4.5.2 嵌套查询4.5.3 分组4.5.4 使用连接4.5.5 分区4.6 小结第5章 超越基本的内存数据查询5.1 常见使用场景5.1.1 查询非泛型集合5.1.2 按照多个条件分组5.1.3 动态查询5.1.4 LINQtoTextFiles5.2 设计模式5.2.1 函数式创建模式5.2.2 ForEach模式5.3 性能分析5.3.1 选择恰当的流操作方式5.3.2 当心立即执行5.3.3 LINQtoObjects会降低代码的性能吗5.3.4 使用LINQtoObjects的代价5.3.5 性能和简洁：鱼和熊掌不可兼得吗5.4 小结第三部分 查询关系型数据第6章 LINQtoSQL入门6.1 走入LINQtoSQL6.1.1 创建对象映射6.1.2 设定DataContext6.2 通过LINQtoSQL读取数据6.3 继续改进查询6.3.1 过滤6.3.2 排序和分组6.3.3 聚集6.3.4 连接6.4 操作有层级关系的对象6.5 数据加载的时机以及为何要关心这些6.5.1 延迟加载6.5.2 立即加载详细信息6.6 更新数据6.7 小结第7章 揭开LINQtoSQL的本质7.1 对象与关系数据的映射7.1.1 通过在类中添加属性7.1.2 使用外部XML文件实现映射7.1.3 使用SqlMetal工具7.1.4 LINQtoSQL设计器7.2 将查询表达式转换为SQL7.2.1 IQueryable接口7.2.2 表达式树7.3 业务实体的生命周期7.3.1 跟踪变化7.3.2 提交变化7.3.3 操作离线数据7.4 小结第8章 LINQtoSQL高级特性8.1 处理并发修改8.1.1 悲观式并发8.1.2 乐观式并发8.1.3 处理并发异常8.1.4 使用事务解决冲突8.2 高级的数据库功能8.2.1 使用SQL语句直接得到结果对象8.2.2 使用存储过程8.2.3 用户自定义函数8.3 改进业务逻辑层8.3.1 预编译的查询8.3.2 使用部分类实现自定义业务逻辑8.3.3 使用部分方法8.3.4 使用对象继承8.4 LINQtoEntities简介8.5 小结第四部分 操作XML第9章 LINQtoXML入门9.1 什么是XMLAPI9.2 为什么需要另外一种XML编程API9.3 LINQtoXML设计原则9.3.1 核心概念：函数式创建9.3.2 核心概念：上下文无关的XML创建过程9.3.3 核心概念：简化了的名称9.4 LINQtoXML中相关类的继承体系9.5 在LINQ中操作XML9.5.1 加载XML9.5.2 解析XML9.5.3 创建XML9.5.4 使用VisualBasicXML字面量创建XML9.5.5 创建XML文档9.5.6 向XML中添加内容9.5.7 从XML中移除内容9.5.8 更新XML的内容9.5.9 操作属性9.5.10 保存XML9.6 小结第10章 使用LINQtoXML查询并转换XML10.1 LINQtoXML轴方法10.1.1 Element10.1.2 Attribute10.1.3 Elements10.1.4 Descendants10.1.5 Ancestors10.1.6 ElementsAfterSelf、NodesAfterSelf、ElementsBeforeSelf和NodesBeforeSelf方法10.1.7 VisualBasic的XML轴属性10.2 标准查询操作符10.2.1 使用Select进行投影10.2.2 使用Where实现过滤10.2.3 排序和分组10.3 使用XPath查询LINQtoXML对象10.4 转换XML10.4.1 LINQtoXML转换10.4.2 使用XSLT转换LINQtoXML对象10.5 小结第11章 LINQtoXML常见场

景11.1 根据XML构建对象11.1.1 目标11.1.2 实现11.2 根据对象创建XML11.2.1 目标11.2.2 实现11.3 根据数据库中的数据创建XML11.3.1 目标11.3.2 实现11.4 过滤并混合从数据库以及XML中得到的数据11.4.1 目标11.4.2 实现11.5 读取XML并更新数据库11.5.1 目标11.5.2 实现11.6 将文本文件转换为XML11.6.1 目标11.6.2 实现11.7 小结第五部分 LINQ无处不在第12章 扩展LINQ12.1 探寻LINQ的扩展机制12.1.1 LINQ各个特性是如何实现的12.1.2 自定义LINQ扩展能够实现什么12.2 创建自定义查询操作符12.2.1 改进标准查询操作符12.2.2 辅助或领域相关的查询操作符12.3 基本查询操作符的自定义实现12.3.1 回顾查询翻译机制12.3.2 查询表达式模式的规范12.3.3 示例1：跟踪标准查询操作符的执行过程12.3.4 限制：查询表达式冲突12.3.5 示例2：非泛型的、领域相关的操作符12.3.6 示例3：非序列的查询操作符12.4 查询Web服务：LINQtoAmazon12.4.1 LINQtoAmazon介绍12.4.2 需求12.4.3 实现12.5 IQueryable和IQueryProvider：LINQtoAmazon高级版本12.5.1 IQueryable和IQueryProvider接口12.5.2 实现12.5.3 究竟发生了什么12.6 小结第13章 应用程序各层中的LINQ13.1 LinqBooks应用程序概览13.1.1 功能13.1.2 程序UI概览13.1.3 数据模型13.2 LINQtoSQL及数据访问层13.2.1 回顾传统的三层架构13.2.2 用专门的数据访问层还是LINQtoSQL13.2.3 LinqBooks中对LINQtoSQL的使用13.3 LINQtoXML的使用13.3.1 从Amazon中导入数据13.3.2 生成RSS13.4 LINQtoDataSet的使用13.5 使用LINQtoObjects13.6 可扩展性13.6.1 自定义查询操作符13.6.2 创建并使用自定义LINQ提供器13.7 展望未来13.7.1 自定义LINQ功能13.7.2 LINQtoXSD：强类型的LINQtoXML13.7.3 PLINQ：支持并行计算的LINQ13.7.4 LINQtoEntities，ADO.NETEntityFramework的LINQ接口13.8 小结第14章 使用LINQ操作DataSet14.1 LINQtoDataSet概览14.2 DataSet回顾14.2.1 DataSet使用场景及功能14.2.2 DataSet是什么14.2.3 VisualStudio2008和.NET3.5为支持LINQtoDataSet进行的改进14.3 查询弱类型DataSet14.3.1 为DataSet加载数据14.3.2 不使用LINQ查询DataSet14.3.3 使用LINQtoDataSet查询弱类型DataSet14.4 查询强类型DataSet14.4.1 生成强类型的DataSet14.4.2 为强类型DataSet加载数据14.4.3 使用LINQtoDataSet查询强类型DataSet14.5 将LINQtoDataSet查询结果绑定至UI控件14.5.1 使用CopyToDataTable将LINQtoDataSet查询结果复制到DataTable14.5.2 使用AsDataView实现双向数据绑定14.6 在LINQtoDataSet中使用查询操作符14.6.1 用于DataRow的Field和SetField操作符14.6.2 集合操作符以及用DataRowComparer比较DataRow14.7 小结附录 标准查询操作符

章节摘录

插图：第1章　LINQ简介本章内容LINQ的起源LINQ的设十目标LINQ to Objects、LINQ to XML和LINQ to SQL初步软件很简单。
它可以归结为两件事情：代码和数据。
开发软件却并非那么简单，其中很重要的一项任务就是编写处理数据的代码。
编写代码时，我们可以选择不同的编程语言。
为应用程序选择开发语言可能需要考虑业务方面的需求、开发者的偏好、开发团队的技能提升、所使用的操作系统以及公司的政策等。
无论选择了哪种语言，在程序开发的某个时候你将不得不开始与数据打交道。
这些数据可能来自于磁盘上的文件、数据库中的表或网络上的XML文档，甚至是上述各项的组合。
归根结底，在任何一个软件项目中，都需要管理数据。
鉴于处理数据的操作在软件开发中无所不在，我们都希望一些功能强大的软件开发平台，如.NET Framework，能够提供一些简单的处理数据方法。
.NET的确对操作数据提供了广泛的支持，不过仍有一些可以改进之处，例如更深层次的语言和数据集成。
这也正是LINO to objects、LINQ to XML和LINQ to SQL的价值所在。

第一图书网, tushu007.com

媒体关注与评论

本书作者是LINQ社区中的重要成员，他们在书中展示了这一技术的精髓……快去享受这一盛宴吧。
——Matt Warren，微软主架构师，LINQ之父LINQ方面的杰作！
深入、全面、客观，令人欲罢不能。
强烈推荐！
——Patrick Smacchia，微软MVP，《C#和.NET 2.0 实战》作者

编辑推荐

《LINQ实战》：微软MVP呕心沥血之作，LINQ圣经，深入全面，引人入胜，丰富实例，兼顾C#和VB.NET语言。
《LINQ实战》内容全面详细，几乎涵盖了所有技术要点。
对于LINQ提供的各种特性，从客观的角度解释其优缺点，使读者能够根据具体情况做出选择。
此外，还讲解了LINQ的可扩展性，可用于更高级的数据处理。
全书组织结构合理。
行文流畅，使读者如同置身于一次技术旅行。
在深入透彻的讲解中恰到好处地运用实例，对同一主题分别给出C#和VB.NIE实例，并在实例的关键点给出注释，真正达到了化繁为简的境界。
LINQ是微软.NET软件开发平台上的一项突破性创新技术，它在面向对象语言与各种数据源之间架设了一座高效的桥梁，大大降低了软件开发的复杂度。

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:http://www.tushu007.com