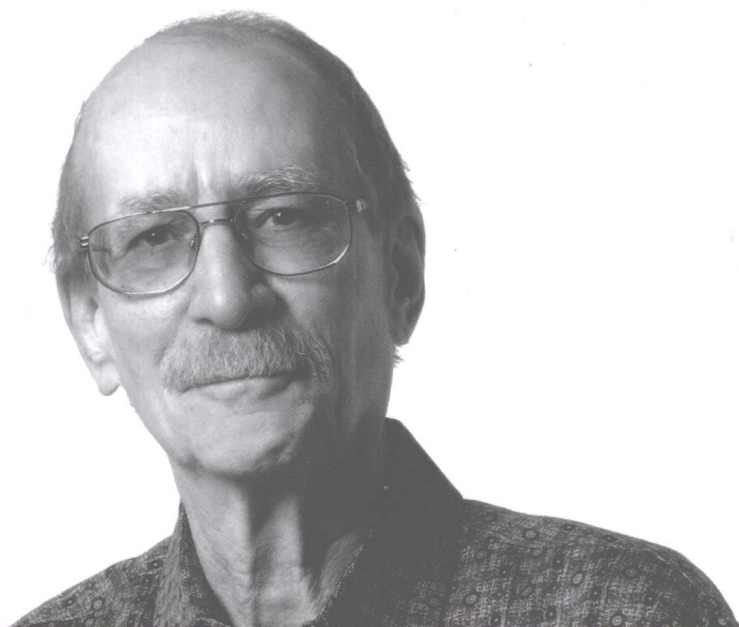


Wrox Programmer to Programmer™



Professional ADO.NET 3.5 with LINQ and the Entity Framework

ADO.NET 3.5

高级编程

——应用 LINQ & Entity Framework

(美) Roger Jennings 著
孟兆炜 译



清华大学出版社

ADO.NET 3.5 高级编程

——应用 LINQ & Entity Framework

(美) Roger Jennings 著
孟兆炜 译

清华大学出版社

北 京

Roger Jennings

Professional ADO.NET 3.5 with LINQ and the Entity Framework

EISBN: 978-0-470-18261-1

Copyright © 2009 by Wiley Publishing, Inc.

All Rights Reserved. This translation published under license.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字：01-2009-2295

本书封面贴有 Wiley 公司防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目(CIP)数据

ADO.NET 3.5 高级编程——应用 LINQ & Entity Framework / (美) 詹宁斯 (Jennings, R.) 著；孟兆炜 译。

—北京：清华大学出版社，2010.6

书名原文：Professional ADO.NET 3.5 with LINQ and the Entity Framework

ISBN 978-7-302-22190-6

I. A… II. ①詹…②孟… III. 计算机网络—程序设计 IV. TP393.09

中国版本图书馆 CIP 数据核字(2010)第 034812 号

责任编辑：王 军 李 阳

装帧设计：孔祥丰

责任校对：成凤进

责任印制：杨 艳

出版发行：清华大学出版社

<http://www.tup.com.cn>

社 总 机：010-62770175

投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈：010-62772015, zhiliang@tup.tsinghua.edu.cn

地 址：北京清华大学学研大厦 A 座

邮 编：100084

邮 购：010-62786544

印 刷 者：北京密云胶印厂

装 订 者：北京鑫海金澳胶印有限公司

经 销：全国新华书店

开 本：185×260 印 张：39.25 字 数：955 千字

版 次：2010 年 6 月第 1 版 印 次：2010 年 6 月第 1 次印刷

印 数：1~3000

定 价：79.00 元

产品编号：030081-01

译者序

LINQ 和实体框架都是 Visual Studio 2008 和 .NET Framework 3.x 推出的最令人激动的特性之一。当前在解决实际问题的过程中，可能遇到的数据种类越来越多，格式越来越复杂。对于每种数据格式，程序员们都要重新学习一套查询语法，这样既费时又费力。LINQ 则在数据库、内存对象、XML、文本等各种类型的数据和面向对象编程语言之间架起了一座桥梁，为各种类型数据的查询提供了一套统一的方式，大大降低了数据访问的复杂度。同时我们知道，不同的数据格式之间存在着阻抗失谐现象(例如，在面向对象编程语言所用的对象模型和关系数据库所用的关系模型之间)，程序员们经常要在解决这个问题上耗费大量精力。而实体框架则在更高抽象层次上提供了一个统一模型，即实体数据模型，从而大大提高了开发效率。

本书详尽介绍了 LINQ 和实体框架的方方面面，并通过丰富的样例程序演示了如何使用这些新技术来提高编程效率。这是关于 LINQ 和实体框架编程方面少见的一本精彩全面而又非常实用的书籍。本书作者 Roger Jennings 是具有二十多年开发经验的资深数据库编程专家，他编写的二十多本编程书籍赢得了读者的广泛好评。我们非常荣幸地把这本译著奉献给广大读者朋友。

本书由孟兆炜翻译，由肖国尊负责监督本书的翻译质量和进度。鉴于译者水平有限，时间仓促，不足和疏漏之处在所难免，敬请广大读者不吝批评指正！读者可以将反馈信息发送到邮箱 wkservice@vip.163.com，我们将不胜感激。

译者

2009年9月

作者简介

Roger Jennings 是一名专业的作者和顾问。他在 Microsoft .NET n 层结构、客户/服务器数据库应用程序、数据密集的基于 Windows 通信基础(Windows Communication Foundation, WCF)的 Web 服务等方面有很深的造诣。从 Professional Extensions for Visual Basic 2.0(代号 Rawhide)开始,他就是 Visual Basic 和 Visual Studio 所有版本的 beta 测试人员。

Roger 先生编写的 25 本计算机方面书籍的印刷量已经超过 125 万册,并被翻译成 20 多种语言。他编写的书籍包括: Wiley/WROX 出版的 *Expert One-on-One Visual Basic 2005 Database Programming*, SAMS 出版的 3 个版本的 *Database Developer's Guide to Visual Basic*, SAMS 出版的两个版本的 *Access Developer's Guide*, QUE 出版的 11 个版本的 *Special Edition Using Microsoft Access*, QUE 出版的两个版本的 *Special Edition Using Windows NT 4.0 Server*。他还编写了多本面向开发者的书籍,内容涉及 Windows 3.1 多媒体、Windows 95、Windows 2000 服务器、活动目录组策略、Visual Basic Web 服务以及微软 Office InfoPath 2003 SP-1 等方面。Roger 先生担任 Redmond 媒体研究组织的 Visual Studio Magazine 杂志及其前身 Visual Basic Programmer's Journal 的特约编辑已经长达 15 年。他也在 Redmond Magazine 上发表文章,并且负责 TechBriefs 专栏以及 Redmond Developer News 的封首语的写作。

Roger 先生有长达 30 年的计算机相关领域经验,他最初从事开发由 Wang 700 型计算器和后继的 Wang BASIC 微计算机所驱动的实时中间数据获取和化学过程控制系统。他是 OakLeaf Systems 公司(一家位于北加利福尼亚的软件顾问公司)的主要开创者,也是 OakLeaf Systems 博客(<http://oakleafblog.blogspot.com>)的作者。他的演示作品 OakLeaf Code of Federal Regulations (CFR) ASP.NET Web Service 荣获微软最佳 .NET 横向解决方案奖(<http://www.microsoft.com/presspass/features/2002/aug02/08-07netwinners.msp>)。

前 言

语言集成查询(Language Integrated Query, LINQ), 以及为支持它而对 C# 3.0 和 VB 9.0 语言所做的扩展, 是 Visual Studio 2008 和 .NET Framework 3.x 最重要的一个新功能。LINQ 是微软首次尝试为内存中各种不同数据集(包括一般对象、关系数据库表中持久化的实体、XML 文档或片段的元素和属性, 以及 RSS 和 Atom syndication feeds 文件等其他多种数据类型)而定义的一种统一查询语言。微软向 Anders Hejlsberg 和他的 C#设计开发团队投入了数百万美元来为 C# 3.0 增添新的功能, 例如 lambda 表达式、匿名类型和扩展方法, 特别增加了对作为语言一部分的 LINQ 标准查询操作符(Standard Query Operators, SQOs)及查询表达式的支持。

VB 9.0 也在 C#开发团队的引导下添加了对应的内容, 值得注意的是, VB 对 LINQ to XML 的实现为语言本身增加了一个新的改进: XML 字面量(literals)。VB 的 LINQ to XML 实现包括了 XML 字面量, 它将会把格式标准的 XML 文档或片段作为 VB 语言的一部分, 而不再需要把元素和属性的名和值从字符串翻译为 XML DOM 结点和值。

本书重在引导读者进行实际的 Windows 和 Web 应用程序的动手开发, 通过 C#和 VB 编程技术的实例演示, 力图使您的 LINQ 技术应用水平得到提高。本书前半部分覆盖的内容包括标准查询操作符 SQO, 以及使用 LINQ 的各种具体实现技术对实现了 IEnumerable<T>和/或 IQueryable<T>接口的集合进行查询。本书后半部分主要讲述 ADO.NET 实体框架、实体数据模型、Entity SQL(eSQL)和 LINQ to Entities。绝大多数代码示例采用了模仿现今常用的数据源, 例如运行于 SQL Server 2005 或者 2008 Express Edition 的 Northwind 样例数据库, 以及根据它的表派生的集合。除第 1 章的样例是简单的命令行项目之外, 代码示例都是 C#和 VB Windows 窗体或者 Web 站点/应用程序。因为仅通过 Hello World 项目这样处理小规模整型数组或姓名数组的 LINQ 查询, 读者是不能够深刻体会到 LINQ 的表现或性能的。

读者对象

本书面向具有 C# .NET 或者 VB.NET 使用经验的开发者中那些希望获得最大幅度提升的人, 他们将因 Visual Studio 2008 中 LINQ(包括 LINQ to Objects、LINQ to SQL、LINQ to DataSets、LINQ to XML)查询处理功能的实现而大大受益。同样, VS 2008 SP1 的实体框架/实体数据模型的对象关系映射、LINQ to Entities 以及越来越多的第三方开源 LINQ 的实现也将同样提升开发者的编程效率和程序性能。

本书假设您基本了解 .NET 2.0 引入的语言功能、Visual Studio 集成开发环境以及关系数据库管理系统(尤其是 Microsoft SQL Server 200X)。具有使用 SQL Server 的 T-SQL 查询

语言以及存储过程的经验将会对阅读本书提供帮助，但这并不是必需的。同样，精通 VS 2005、.NET 2.0、C# 2.0 或者 VB 8.0 将会有助于对本书的 C# 3.0 或者 VB 9.0 代码示例的了解，但这也并不是必要条件。

注释：

微软的 .NET 代码样例主要用 C# 3.0 编写。本书章节和示例项目中的所有代码样例均有 C# 或者 VB 版本，但 T-SQL 或 JavaScript 程序除外。

本书内容

本书主要讲述如下几方面的编程技术：基于 LINQ to Objects 的 System.Linq 和 System.Linq.Expressions 名称空间，基于 LINQ to SQL 的 System.Data.Linq 名称空间，基于 LINQ to DataSet 的 System.Data.Linq 名称空间，基于 LINQ to XML 的 System.Xml.Linq 名称空间，以及基于实体框架 Entity SQL 的 System.Data.Entity 和 System.Web.Entity 名称空间。

组织方式

本书分为 5 部分，每部分各有 1~5 章。除了第 I 部分之外，每个部分都有一个大约 4 页的简介，对本部分各章的内容作一概述并介绍相关背景。绝大多数章节建立在先前章节知识的基础上，所以应该循序渐进地阅读。下面将对各部分和其所属各章内容作一些简要描述。

第 I 部分 ADO.NET 3.5 概览

第 I 部分只有一章，通过实例概述了 LINQ 和它在 4 个主要领域(LINQ to SQL、LINQ to Datasets、LINQ to XML、LINQ to Entities)的实现。本章的 26 个样例项目演示了如何利用 LINQ 编码技术并基于 C# 3.0 和 VB 9.0 语言实现简单控制台程序、Windows 窗体和 Web 站点应用程序。

- **第 1 章** “使用 ADO.NET 提供的新方法访问数据”用简单的 C# 和 VB 代码例子演示了如何使用 LINQ to Objects 技术实现对内存中对象的查询，以及如何和由 LINQ 填充的通用 List<T>集合进行数据绑定，使用 LINQ to SQL 进行对象/关系映射，使用 LINQ to Datasets 连接 DataTables，使用 LINQ to Entities 创建 EntitySets，使用 LINQ to XML 查询和操作 XML InfoSets，使用 LINQ to XSD 对强类型 XML 文档执行查询。本章结尾处对实体框架和实体数据模型进行了介绍。

第II部分 LINQ 简介

第II部分中的各章解释了.NET 3.5及C# 3.0和VB 9.0语言为支持LINQ对内存对象的查询所做的改进。LINQ to Objects是所有其他具体LINQ实现的基础。

- 第2章“理解LINQ的体系结构和实现”首先介绍为支持LINQ而对C#和VB语言进行的扩展以及名称空间、LINQ标准查询操作符、表达式树和编译查询，并对领域相关的LINQ实现进行预习。本章的C#和VB样例项演示了对象、数组、集合初始化程序、扩展方法、匿名类型、谓词、Lambda表达式及简单的查询表达式。
- 第3章“使用LINQ to Objects执行LINQ查询表达式”将50个标准查询操作符划分为如下分组：限制操作符、投影操作符、分区操作符、连接操作符、排序操作符、分组操作符、集合操作符、转换操作符和相等操作符，并且列出了它们在C#和VB中的关键字。VS 2008 SP1包括了LINQ Project Sample Query Explorer的C#和VB版本，但是这两个版本的浏览器并没有使用现实世界的集合作为数据源。因此，该章讲述了LINQ内存对象生成器(LINQ in-memory object generator, LIMOG)，这个工具软件可以为(比LINQ Project Sample Query Explorer中使用的更为复杂的)典型业务对象生成C# 3.0或VB 9.0类声明。使用这些业务对象作为数据源的样例C#和VB查询比使用整数或姓名数组要更具表现力。
- 第4章“使用高级查询操作符和表达式”介绍了在实体拥有关联实体的情况下，如何对对象图执行LINQ查询。该章首先给出聚集操作符的例子，解释了Let临时局部变量操作符的使用，展示了如何使用Group By操作符进行聚集查询，完成左外连接的等价操作，并利用Contains()标准查询操作符模拟SQL的IN()函数。您将学习如何通过编译查询来提高性能，并在没有对关系持久性存储进行查询的开销的情况下创建mock测试对象类。

第III部分 使用领域相关的LINQ实现

第III部分各章将引导读者进入LINQ应用程序学习的下一个阶段，即SQL Server 200x数据库的领域相关的版本、使用DataTable、管理XML Infosets，以及正在快速扩展的第三方LINQ实现队伍中的一些成员。每章为了讲解其中的内容都提供了C#和VB样例项目。

- 第5章“使用LINQ to SQL和LinqDataSource”介绍了微软第一个达到产品发布状态的O/RM工具LINQ to SQL，并讲述了如何使用图形化的O/R Designer或命令行SqlMetal.exe为实体类型自动生成类文件。该章也介绍了怎样使用设计器或XML Editor来编辑*.dbml映射文件，如何实例化DataContext对象，以及如何使用LINQ to SQL作为T-SQL查询或存储过程的数据访问层。最后介绍了如何在Web站点或应用程序中使用ASP.NET LinqDataSource控件。

- 第 6 章“使用 LINQ to DataSet 查询 DataTable”首先比较了 DataSet 和 DataContext 对象及其功能，并讲述了 DataSetExtensions 的功能。然后介绍了如何查询无类型和类型化的 DataSet、创建查找列表，以及如何使用 AsDataView() 方法为数据绑定生成 LinqDataView。本章最后介绍了怎样将 LINQ 查询结果复制到 DataTable。
- 第 7 章“使用 LINQ to XML 处理文档”描述了 LINQ 最强大的一个功能：管理 XML Infosets。该章演示了 LINQ to XML 所具有的和 XQuery 1.0 及 XPath 2.0 等同甚至超越它们的查询和浏览能力。该章也证明 LINQ to XML 文档转换在绝大部分常见情形下可以替代 XQuery 和 XSLT 1.0+。您将学习到怎样使用 VB 9.0 的 XML 字面量来构造 XML 文档、使用 GroupJoin() 来生成层次化文档以及如何在 C# 和 VB 中使用 XML 名称空间。
- 第 8 章“探索第三方和新兴的 LINQ 实现”描述了可以在 LINQ to Objects 查询中充分利用多核 CPU 性能的微软 Parallel LINQ(也称为 PLINQ)、如何将 LINQ 查询翻译为 REST(具象状态传输)URL 的 LINQ to REST(它使用 HTML GET、POST、PUT 和 DELETE 方法定义到 Web 服务的请求)以及由 Bart de Smet 编写的 LINQ to Active Directory 和 LINQ to SharePoint 的第三方实现。

第IV部分 ADO.NET 实体框架介绍

第IV部分深入介绍了实体数据模型(EDM)的体系结构和它的第一个实体框架对象/关系映射工具的具体实现。微软希望实体数据模型成为定义概念数据模型的通用规范，在该模型中来自各种各样的数据源的数据都被表示为实体(entity)和其关联(association)(也被称为关系)。除了用于持久化对象的关系数据库之外，实体数据模型也将成为未来用于数据同步和报表服务的组件之一。

- 第 9 章“使用实体数据模型提升数据抽象级别”首先介绍了实体数据模型以及作为对象/关系映射工具和 ADO.NET DataSet 继承者的实体框架的发展进程，并概要描述了实体关系数据模型和图，然后详细分析了实体框架体系结构。随后简介了 Entity SQL(eSQL)语言、eSQL 查询、客户视图和 Object Service(包括ObjectContext、MetadataWorkspace 和 ObjectStateManager)。第 11 章详细描述了 eSQL 和这些对象。该章的 eSQL 查询和 Object Service 样例代码被扩充成两个 C# 和 VB 样例项目。
- 第 10 章“定义存储层、概念层和映射层”深入分析了在运行时生成*.ssdl(存储架构定义语言)、*.msl(映射规范语言)和*.csdl 文件的*.edmx 文件的结构。您将了解在图形化 EDM Designer 不能解决问题的情况下怎样手动编辑*.edmx 文件，以及怎样实现每个层次结构一个表(TPH)继承模型和通过遍历 MetadataWorkspace 获取属性值。该章的 4 个样例项目分别演示了映射、用存储过程代替查询和 TPH 继承。
- 第 11 章“Entity SQL 简介”分析了实体框架的新 eSQL 方言，eSQL 添加了用于区分查询实体和查询关系表的关键字。您将会学习到怎样使用 Zlatko Michaelov 的

eBlast 工具编写和分析 eSQL 查询, 然后进一步讨论了 eSQL 和 T-SQL SELECT 查询的区别(eSQL 第一版并不支持 INSERT、UPDATE、DELETE 和其他 SQL 数据操作语言结构)。该章讲述了如何对 EntityClient 执行 eSQL 查询、测量 eSQL 相对于 T-SQL 的性能损失、执行参数化 eSQL 查询以及如何使用 SQL Server Compact 3.5 作为数据存储区。C#和 VB 样例项目演示了该章所讲述的编程技术。

第 V 部分 实现 ADO.NET 实体框架

第 V 部分从实战角度讲述在 SQL Server 200x 中如何使用实体框架实现对象/关系映射、使用 LINQ to Entities 查询、使用 LINQ to Entities 更新传统实体和具有复杂类型的实体、将对象服务的ObjectContext 数据绑定到 Windows 窗体和 ASP.NET 服务器控件, 以及如何利用实体框架作为其他数据感知的 .NET 技术的数据源。每章都包含由该章的代码例子扩充而来的 C#和 VB 样例项目。

- 第 12 章“利用对象服务和 LINQ to Entities”专门讨论了如何操作对象服务 API 的ObjectContext。该章随后演示了 *ModelNameEntities* 和 *EntityName* 对象的局部类的使用、执行 eSQL ObjectQuery、相关实体的延迟加载或预先加载以及相关实体的排序和过滤。该章最后讲述了如何通过组合 QueryBuilder 方法来编写 ObjectQuery、LINQ to Entities 查询以及参数化 ObjectQuery。
- 第 13 章“更新实体和复杂类型”阐述了如何对 EntitySet 执行创建、更新和删除(CUD)操作, 以及管理并发冲突。这一章详细介绍了 ObjectContext.ObjectStateManager 及其子对象, 它们负责基于 EntityKey 唯一识别对象以及执行对象更改跟踪操作。本章也介绍了如何进行创建和更新操作的验证、使用存储过程完成更新, 以及使用复杂类型。
- 第 14 章“将实体绑定到数据感知控件”讲述了如何基于 ObjectContext.EntitySet 实例创建设计时数据源, 并通过拖放操作在 Windows 窗体中添加 BindingNavigator、BindingSource、绑定的 TextBox 及 DataGridView。您也将学习到怎样在 DataGridView 控件的 ComboBox 列中更新 EntityReference 和 EntitySet 值(不能直接更新 EntitySet 值, 必须首先删除原有值, 然后才能添加一个具有所需值的新成员)。本章结尾演示了绑定到 GridView 和 DropDownList 控件的 ASP.NET EntityDataSource 控件的使用。
- 第 15 章“使用实体框架作为数据源”主要讲述了如何使用实体框架作为 ADO.NET 数据服务框架(之前的代号“Project Astoria”仍适用)的数据源, 这也是将实体框架第一版部署为 Web 服务提供者的推荐方法(实体框架第二版将支持直接使用 WCF 进行 n 层数据访问)。该章给出的 Windows 窗体示例使用 Astoria 的 .NET 3.5 客户端库并按照 Atom 出版(AtomPub 或 APP)传输格式来显示和更新实体实例。而 Web 窗体项目则使用 AJAX 客户端库和 JSON 作为传输格式。

源代码

在读者学习本书中的示例时，可以手工输入所有代码，也可以使用本书附带的源代码文件。本书使用的所有源代码都可以从本书合作站点 <http://www.wrox.com/> 或 www.tupwk.com.cn/downpage 上下载。登录到站点 <http://www.wrox.com/>，使用 Search 工具或使用书名列表就可以找到本书。接着单击本书细目页面上的 Download Code 链接，就可以获得所有的源代码。

注释：

由于许多图书的标题都很类似，所以按 ISBN 搜索是最简单的，本书英文版的 ISBN 是 978-0-470-18261-1。

在下载了代码后，只需用自己喜欢的解压缩软件对它进行解压缩即可。另外，也可以进入 <http://www.wrox.com/dynamic/books/download.aspx> 上的 Wrox 代码下载主页，查看本书和其他 Wrox 图书的所有代码。

勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误，但是错误总是难免的，如果您在本书中找到了错误，例如拼写错误或代码错误，请告诉我们，我们将非常感激。通过勘误表，可以让其他读者避免受挫，当然，这还有助于提供更高质量的信息。

请给 wkservice@vip.163.com 发电子邮件，我们就会检查您的反馈信息，如果是正确的，我们将在本书的后续版本中采用。

要在网站上找到本书英文版的勘误表，可以登录 <http://www.wrox.com>，通过 Search 框或书名列表查找本书，然后在本书的细目页面上单击 Book Errata 链接。在这个页面上可以查看到 Wrox 编辑已提交和粘贴的所有勘误项。完整的图书列表还包括每本书的勘误表，网址是 www.wrox.com/misc-pages/booklist.shtml。

P2P.WROX.COM

要与作者和同行讨论，请加入 p2p.wrox.com 上的 P2P 论坛。这个论坛是一个基于 Web 的系统，便于您张贴与 Wrox 图书相关的消息和相关技术，与其他读者和技术用户交流心得。该论坛提供了订阅功能，当论坛上有新的消息时，它可以通过电子邮件给您传送感兴趣的论题。Wrox 作者、编辑和其他业界专家和读者都会到这个论坛上来探讨问题。

在 <http://p2p.wrox.com> 上，有许多不同的论坛，它们不仅有助于阅读本书，还有助于开发自己的应用程序。要加入论坛，可以遵循下面的步骤：

- (1) 进入 p2p.wrox.com，单击 **Register** 链接。
- (2) 阅读使用协议，并单击 **Agree** 按钮。
- (3) 填写加入该论坛所需要的信息和自己希望提供的其他信息，单击 **Submit** 按钮。
- (4) 您会收到一封电子邮件，其中的信息描述了如何验证账户，完成加入过程。

注释：

不加入 P2P 也可以阅读论坛上的消息，但要张贴自己的消息，就必须加入该论坛。

加入论坛后，就可以张贴新消息，响应其他用户张贴的消息。可以随时在 **Web** 上阅读消息。如果要让该网站给自己发送特定论坛中的消息，可以单击论坛列表中该论坛名旁边的 **Subscribe to this Forum** 图标。

关于使用 **Wrox P2P** 的更多信息，可阅读 **P2P FAQ**，了解论坛软件的工作情况以及 **P2P** 和 **Wrox** 图书的许多常见问题。要阅读 **FAQ**，可以在任意 **P2P** 页面上单击 **FAQ** 链接。

目 录

第 I 部分 ADO.NET 3.5 概览	
第 1 章 使用 ADO.NET 3.5 提供的新方法访问数据3	
1.1 语言集成查询 LINQ5
1.1.1 LINQ to Objects8
1.1.2 LINQ to SQL14
1.1.3 DataContext 编程22
1.1.4 LINQ to Datasets34
1.1.5 LINQ to Entities38
1.1.6 LINQ to XML39
1.1.7 LINQ to XSD48
1.2 ADO.NET 实体框架和实体数据模型49
1.2.1 从物理架构到概念架构的映射51
1.2.2 使用实体数据模型向导来创建默认实体数据模型53
1.2.3 使用实体数据模型 Designer 来修改存储架构到概念架构的映射54
1.2.4 根据数据模型创建和绑定数据源56
1.2.5 具体化对象上下文57
1.3 小结58
第 II 部分 LINQ 简介	
第 2 章 理解 LINQ 的体系结构和实现 ...65	
2.1 .NET Fx 3.5 中支持 LINQ 的名称空间66
2.2 支持 LINQ 的 C#和 VB 扩展67
2.2.1 隐式类型化局部变量68
2.2.2 对象初始化器70
2.2.3 使用对象初始化器的数组初始化器71
2.2.4 集合初始化器72
2.2.5 匿名类型72
2.2.6 扩展方法73
2.2.7 匿名方法和通用谓词76
2.2.8 Lambda 表达式78
2.2.9 标准查询操作符81
2.2.10 表达式树和编译查询84
2.2.11 IQueryable<T>接口和领域相关的 LINQ 实现89
2.3 小结89
第 3 章 使用 LINQ to Objects 执行 LINQ 查询表达式91	
3.1 标准查询操作符分组92
3.2 C# 3.0 和 VB 9.0 中作为关键字的标准查询操作符93
3.3 LINQ 项目样例查询浏览器94
3.4 LINQ to Objects 代码示例的样例类96
3.4.1 C#类定义和初始化代码样例97
3.4.2 VB 类定义和初始化代码样例99
3.5 限制操作符: Where100
3.5.1 简单 Where 表达式101
3.5.2 复合 Where 表达式101

3.5.3 带有 Index 参数的方法 调用及 IndexOf() 的使用····· 103	3.15 元素操作符····· 144
3.6 投影操作符····· 105	3.15.1 First、FirstOrDefault····· 145
3.6.1 Select····· 105	3.15.2 Last、LastOrDefault····· 145
3.6.2 SelectMany····· 107	3.15.3 Single、SingleOrDefault····· 146
3.7 分区操作符····· 110	3.15.4 DefaultIfEmpty····· 146
3.7.1 Take····· 111	3.15.5 ElementAt、 ElementAtOrDefault····· 147
3.7.2 Skip····· 111	3.16 生成操作符····· 147
3.7.3 Skip/Take 样例····· 112	3.16.1 Range····· 148
3.7.4 TakeWhile····· 113	3.16.2 Repeat····· 148
3.7.5 SkipWhile····· 113	3.16.3 Empty····· 148
3.7.6 SkipWhile/TakeWhile 样例····· 114	3.17 量词操作符····· 148
3.8 连接操作符····· 114	3.17.1 Any····· 149
3.8.1 Join····· 115	3.17.2 All····· 149
3.8.2 GroupJoin····· 117	3.17.3 Contains····· 150
3.9 串联操作符: Concat····· 119	3.18 聚集操作符····· 150
3.10 排序操作符····· 121	3.18.1 Count 和 LongCount····· 150
3.10.1 OrderByDescending····· 122	3.18.2 Min、Max、Sum 和 Average····· 151
3.10.2 ThenBy····· 122	3.18.3 Aggregate····· 152
3.10.3 ThenByDescending····· 122	3.19 小结····· 154
3.10.4 Reverse····· 123	
3.10.5 Ordering 操作符的示例····· 123	第 4 章 使用高级查询操作符和 表达式····· 157
3.11 分组操作符: GroupBy····· 125	4.1 浏览聚集操作符的基本 查询语法····· 157
3.11.1 使用方法调用语法 的 GroupBy 操作····· 126	4.1.1 数字操作符的基本方法 调用语法····· 158
3.11.2 使用查询表达式语法 的 GroupBy 操作····· 127	4.1.2 使用 Let 临时局部 聚集变量····· 160
3.12 集合操作符····· 129	4.1.3 格式化查询输出····· 161
3.12.1 Distinct····· 129	4.2 在聚集查询中使用 Group By····· 162
3.12.2 Union····· 130	4.2.1 分组关联子对象····· 163
3.12.3 Intersect····· 131	4.2.2 分组连接器对象····· 164
3.12.4 Except····· 132	4.2.3 用层次化分组连接表达式 组合连接操作和 Group By 操作····· 166
3.13 转换操作符····· 133	4.2.4 使用嵌套 LINQ 查询比较 组连接····· 168
3.13.1 AsEnumerable····· 133	
3.13.2 AsQueryable····· 133	
3.13.3 Cast····· 135	
3.13.4 OfType····· 137	
3.13.5 To... 操作符····· 138	
3.14 相等操作符: SequenceEqual····· 143	

4.2.5 使用实体关联模拟 左外连接.....	170	5.2.1 LINQ to SQL 查询管道.....	214
4.3 充分利用 Contains()标准 查询操作符.....	171	5.2.2 添加、更新和删除对象.....	217
4.3.1 使用复合 OR 操作符模拟 SQL WHERE 子句.....	171	5.2.3 检测并解决并发冲突.....	221
4.3.2 使用 Contains()模拟 SQL IN()函数.....	173	5.2.4 用存储过程代替动态 SQL.....	224
4.4 通过编译查询表达式树 提高性能.....	174	5.2.5 将 LINQ to SQL 变为 中间层.....	227
4.5 测试 LINQ to SQL 和 LINQ to Entities 项目的 Mocking 集合.....	178	5.3 使用 LinqDataSource 控件 进行 ASP.NET 数据绑定.....	228
4.5.1 创建 Mock 对象和 初始化器.....	179	5.3.1 在页面中加入 LinqDataSource.....	229
4.5.2 使用 GroupJoin 表达式 创建对象图.....	184	5.3.2 用 EntityRef 代替数据 绑定 Web 控件的外键值.....	230
4.6 小结.....	189	5.3.3 预先加载 EntityRef 值以 降低数据库服务器流量.....	232
 		5.4 将 Windows 窗体控件绑定 到实体.....	235
第III部分 使用领域相关的 LINQ 实现		5.4.1 自动生成必要的层次化 数据编辑窗体.....	235
 		5.4.2 持久化对实体的编辑和 集合的更改.....	237
第5章 使用 LINQ to SQL 和 LinqDataSource		5.5 小结.....	243
 		第6章 使用 LINQ to DataSet 查询 DataTable	245
5.1 使用 LINQ to SQL 进行 对象关系映射.....	199	6.1 比较 DataSets 和 DataContexts.....	246
5.1.1 使用 LINQ to SQL 对象关系 设计器将表映射到实体集.....	199	6.2 探索 LINQ to DataSet 的 各种功能.....	247
5.1.2 使用 SqlMetal.exe 生成部分 实体类和映射文件.....	201	6.3 运行只读的 LINQ to DataSet 查询.....	249
5.1.3 使用 *.dbml 和 *.xml 映射 文件.....	204	6.3.1 查询无类型的 DataSet.....	250
5.1.4 分析生成类.....	207	6.3.2 定制查找列表.....	255
5.1.5 初始化 DataContext 及其 对象图.....	211	6.3.3 查询类型化 DataSet.....	256
5.2 使用 LINQ to SQL 作为数据 访问层.....	213	6.4 使用 AsDataView()为数据 绑定创建 LinqDataViews.....	259
		6.5 将 LINQ 查询结果复制 到 DataTable.....	263
		6.5.1 复制类型化的 DataRow.....	264
		6.5.2 基于投影处理匿名类型.....	265

6.6	小结	269
第 7 章 使用 LINQ to XML		
	处理文档	271
7.1	将 XML 集成到 CLR	271
7.1.1	使用 Xsl 最小化 XML/ 对象的失配	272
7.1.2	使用 C# 查询 XML	272
7.2	System.Xml.Linq 名称空间	273
7.3	查询基本的 XML Infoset	275
7.3.1	为 VB 查询推断架构和 启用智能感知技术	277
7.3.2	利用 VB 9.0 的 Axis 属性	278
7.3.3	元素和属性内容的隐式 键入与显式键入	280
7.4	组合 XML Infosets	282
7.4.1	在 C# 3.0 中使用功能构造	284
7.4.2	使用 VB 9.0 进行常量 XML 构造	287
7.5	业务文档中的分组元素和 聚集数量值	291
7.5.1	使用 GroupJoin 生成 层次化文档	291
7.5.2	使用一对多和多对一关联	293
7.5.3	聚集每个客户的 Order_Details 和 Orders 小计	295
7.6	使用 XML 名称空间和 局部名称	298
7.6.1	C# 3.0 中的 XML 名称空间	300
7.6.2	VB 9.0 中的 XML 名称空间	305
7.7	执行异构连接和查找操作	310
7.7.1	使用查找操作添加子 元素组	310
7.7.2	将文档连接至插入元素	315
7.7.3	将文档和 LINQ to SQL 或者 LINQ to Object 实体相连接	316

7.8	小结	318
第 8 章 探索第三方和新兴的 LINQ 实现		
8.1	新兴的微软 LINQ 实现	322
8.1.1	Parallel LINQ	322
8.1.2	LINQ to REST	329
8.1.3	LINQ to XSD	342
8.1.4	LINQ to Stored XML	343
8.2	第三方领域相关的 LINQ 实现	344
8.2.1	LINQ to Active Directory	344
8.2.2	LINQ to SharePoint	345
8.3	小结	351

第 IV 部分 ADO.NET 实体框架介绍

第 9 章 使用实体数据模型提升 数据抽象级别		
9.1	理解实体关系模型	361
9.1.1	实体关系和实体数据 模型术语	362
9.1.2	实体关系图	363
9.2	理解实体框架结构和组件	364
9.2.1	利用实体数据模型设计器 实现从物理层到概念层的 映射	366
9.2.2	分析 ModelName.edmx 文件的节	371
9.3	使用 Entity Client、Entity SQL 和 Client Views	377
9.3.1	用 Entity SQL 编写 EntityQueries	378
9.3.2	作为客户视图执行 Entity SQL 查询	378
9.4	使用 Object Service	383
9.4.1	使用ObjectContext	384
9.4.2	使用 Entity SQL 编写 ObjectQueries	386

9.4.3 使用查询生成器方法合成 ObjectQuery.....	389	11.2.8 用 REF、DEREF、 CREATEREF、ROW 和 KEY 管理实体引用	445
9.4.4 使用 LINQ to Entities 提供者	391	11.2.9 使用类型构造函数 创建 ROW、Multiset 和 EntityType 的实例	447
9.5 理解关于持久化透明的争论	392	11.2.10 UNION、INTERSECT、 OVERLAPS 和 EXCEPT 集合操作需要子查询	449
9.6 小结	394	11.2.11 对集合操作符返回的集 合排序需要嵌套查询	449
第 10 章 定义存储层、概念层 和映射层	395	11.2.12 用于集合的操作符 ANYELEMENT 和 FLATTEN	450
10.1 探索并定制 EDMX 文件	397	11.2.13 ORDER BY 子句的 SKIP 和 LIMIT 子句 处理分页	451
10.1.1 存储模型(SSDL 内容)	399	11.2.14 IS OF、OFTYPE 和 TREAT 是多态查询的 类型操作符	452
10.1.2 概念模型(CSDL 内容)	412	11.2.15 在 WHERE 子句的约束 中使用返回聚集值的子 查询将导致抛出异常	453
10.1.3 映射(MSL 内容)	415	11.3 对 EntityClient 执行 eSQL 查询	454
10.2 实现 TPH 继承	419	11.3.1 解析 EntityDataReader 返回 的 IExtendedDataRecord	455
10.2.1 设定鉴别器列并创建 派生类	420	11.3.2 测量 EntitySQL 查询的 性能损失	458
10.2.2 查询基类和派生类	422	11.3.3 执行参数化 eSQL 查询	459
10.3 遍历 MetadataWorkspace	428	11.4 使用 SQL Server Compact 作为实体框架的数据存储	461
10.4 小结	431	11.5 小结	462
第 11 章 Entity SQL 简介	433	第 V 部分 实现 ADO.NET 实体框架	
11.1 使用 eSqlBlast Entity SQL 查询工具	434	第 12 章 利用对象服务和 LINQ to Entities	469
11.1.1 连接页面	434	12.1 探索生成的实体类	469
11.1.2 模型页面	435		
11.1.3 查询页面	435		
11.1.4 结果页面	436		
11.2 理解 Entity SQL 和 Transact-SQL 的不同	438		
11.2.1 强制使用实体别名前缀	439		
11.2.2 要求使用显式投影	440		
11.2.3 VALUE 修饰符平面化 结果	440		
11.2.4 使用圆点标记符语法 返回多对一导航属性	441		
11.2.5 一对多导航属性需要 嵌套查询	442		
11.2.6 连接是最后的手段	443		
11.2.7 NAVIGATE 是圆点 表示法或嵌套查询的 一个复杂的替代候选	444		