Visual Studio.NET 2008 开发一册通

LINQ

从基础到项目实战

卓文华讯　解本巨　李宗颜　宫生文　编著

本书特色

- 对知识点的讲述通俗易懂，深入浅出，并融入了作者多年的开发心得
- 内容全面，重点突出，对LINQ中的疑点、难点分析透彻
- 实战项目的选择和设计独具匠心，实战项目的讲解采用现代软件工程方法作指导
- 小实例的讲述分割合理，格式醒目

全书提供了161个实例和1个综合应用实例，非常实用

代码及PPT下载：www.cip.com.cn　www.ecainfo.com

化学工业出版社

Visual Studio.NET 2008 开发一册通

LINQ

从基础到项目实战

卓文华讯　解本巨　李宗颜　宫生文　编著

化学工业出版社
·北　京·

LINQ（Language Integrated Query）是.NET Framework 3.5 推出的全新数据访问技术。借助于 LINQ 技术，我们可以使用一种类似 SQL 的语法来查询任何形式的数据。本书是一本学习 LINQ 基本概念和基础理论、掌握 LINQ 开发技术的实用参考书，较为全面地介绍了 LINQ 语法及其相关技术。本书在详细介绍 LINQ 语法及其相关技术的同时，尤为注重 LINQ 技术在实践方面的应用。

本书采用理论与实例结合、相互渗透、逐步引导的方法，通过实例剖析技术的具体应用，使读者能较快地入门并逐步精通。

本书可供相关技术研究人员、广大.NET 应用程序开发者和用户参考，也可作为高等院校计算机、电子商务以及信息类相关专业的教材。

图书在版编目（CIP）数据

LINQ 从基础到项目实战 / 解本巨，李宗颜，宫生文编著. —北京：化学工业出版社，2010.2
（Visual Studio.NET 2008 开发一册通）
ISBN 978-7-122-07302-0

Ⅰ. L… Ⅱ. ①解… ②李… ③宫… Ⅲ. 计算机网络-程序设计 Ⅳ. TP393

中国版本图书馆 CIP 数据核字（2009）第 232080 号

责任编辑：陈 静　　装帧设计：尹琳琳
责任校对：洪雅姝

出版发行：化学工业出版社（北京市东城区青年湖南街 13 号 邮政编码 100011）
印　　装：北京白帆印务有限公司
787mm×1092mm 1/16 印张 25 字数 590 千字 2010 年 2 月北京第 1 版第 1 次印刷

购书咨询：010-64518888（传真：010-64519686） 售后服务：010-64518899
网　　址：http://www.cip.com.cn
凡购买本书，如有缺损质量问题，本社销售中心负责调换。

定　　价：45.00 元

前　言

微软公司在 2008 年正式发布了 Visual Studio 2008 和.NET Framework 3.5 版本，对.NET 做了很多重大的改进，其中 LINQ（Language Integrated Query）技术尤为耀眼。LINQ 是.NET Framework 3.5 推出的全新数据访问技术。借助于 LINQ 技术，我们可以使用一种类似 SQL 的语法来查询任何形式的数据。

LINQ 的多项革命性特征必将引领信息处理技术走向新的高度，所以众多读者期望能更加深入、透彻地了解 LINQ 所带来的各种编程优势，从而对其自身开发工作起到促进作用。本书理论联系实际，通过讲解使用 LINQ 技术在解决开发过程中经常遇到的常见问题，使读者更好地理解 LINQ 技术。

本书特色

（1）对知识点的讲解通俗易懂，深入浅出，并无缝地融入了编者多年的开发心得。

编者具有在中外知名软件企业从事一线开发的经历，具有多年的 C#及 LINQ 开发经验，对编程中所需的 LINQ 知识点有独特的见解，并能用通俗易懂的语言，深入浅出地表达出来。

（2）内容全面，重点突出，对 LINQ 中的疑点、难点分析透彻。

编者曾多次为在校本科生和软件培训机构讲授 C#及 LINQ 课程，因此既对书中的重点内容有较好的把握，也对读者在学习中可能会碰到的疑点、难点有深刻的了解。书中每个重要模块及重要的知识点均会以“专家讲解”的形式来更好地提醒和指导读者。

（3）实战项目的选择和设计独具匠心，其讲解采用现代软件工程方法做指导。

选取了一个具有 LINQ 代表性的实战项目，重点讲述了实际开发中 LINQ 技术的应用特性，全过程讲解实际项目开发中如何进行需求分析、系统设计、数据库设计和编码等。

（4）小实例的讲述分割合理，格式醒目。

每个小实例分割为“代码演示”、“专家讲解”和“结果验证”3 部分，每一部分都有特别而又统一的区别于正文的格式，如字体、字号等，从而使“枯燥”的大段代码变得轻松活泼，更有利于读者阅读和理解。

本书内容

本书是一本学习 LINQ 基本概念和基础理论、掌握 LINQ 开发技术的实用参考书，较为全面地介绍了 LINQ 的语法及其相关技术，并尤为注重 LINQ 技术在实践方面的应用。全书共分为 6 章，具体内容如下。

第 1 章首先介绍了什么是 LINQ、LINQ 的设计目标及 LINQ 产生的背景；接着介绍了 LINQ 的分类，分别介绍了 LINQ to OBJECT、LINQ to ADO.NET 和 LINQ to XML 的概念等内容；最后讲解了 3 个分别使用 LINQ to OBJECT、LINQ to SQL 和 LINQ to XML 的简单实例。

第 2 章首先介绍了 C# 2.0 语言中的一些重要概念，如泛型、委托、匿名方法和列举以及 yield 关键字，这些概念对于理解 C# 3.0 语言的新特性是非常重要的；接下来结合实例仔细讲解了 C# 3.0 语言的新特性，包括局部变量类型推断、Lambda 表达式、扩展方法、对象初始化表达式和匿名类型等。

第 3 章介绍了 LINQ to OBJECT 编程接口，首先从基础知识讲起，讲述了 LINQ to OBJECT 编程接口的几个重要概念、接口等；接下来结合大量实例，重点讲解 LINQ to OBJECT 编程接口的大量标准查询操作符；最后讲解了如何在 ASP.NET 和 WinForm 中使用 LINQ to OBJECT。

第 4 章介绍了 LINQ to SQL 编程接口，首先介绍了 ORM 框架等内容；接下来介绍了在 LINQ to SQL 中如何建立对象-关系映射，讲解了 4 种建立映射的方法；再接下来讲解了 LINQ to SQL 编程接口中最重要的对象——DataContex 对象；然后结合实例讲解使用 LINQ to SQL 进行数据库操作的方法，包括插入、查询、更新和删除操作等；最后介绍了并发访问冲突检测与处理方法。

第 5 章介绍了 LINQ to XML 编程接口，首先从基础知识讲起，介绍了 LINQ to XML 编程接口相比现有 XML 编程接口的优点；接下来结合大量实例，重点讲解 LINQ to XML 编程接口的类层次结构；最后讲解了常用的 LINQ to XML 操作。

第 6 章介绍了一个应用 LINQ 技术的综合实例，分别从系统功能、数据库设计以及页面设计等方面进行了讲解。该实例中运用了大量的 LINQ 特性，通过学习本章，读者可以将前面几章内容融会贯通，真正掌握 LINQ 实战技巧。

本书技术阐述与实践应用相结合，强调理论联系实际开发需求。书中的应用实例均来自于实际开发，读者可以对其稍加修改后直接应用。

本书由卓文华讯解本巨、李宗颜、宫生文编著，并在编写过程中，得到多位专家、教授的指导。由于 LINQ 相关技术标准的不断更新，加之时间仓促和编者水平有限，本书的内容难免会有疏漏和不足之处，恳请广大读者和同行批评指正。我们的联系邮箱是：qditpub@gmail.com。

编　者

2009 年 12 月

目　录

CONTENTS

CONTENTS

第 1 章 LINQ 概述

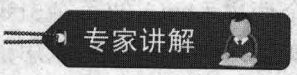
本章简单介绍 LINQ 的概念、LINQ 的设计目标以及 LINQ 的分类，通过 3 个使用 LINQ 的实例引领读者开始 LINQ 的学习。本章使用的实例代码运用了大量 C# 3.0 语言的新特性，如对象和集合初始化器以及扩展方法等。读者在学习本章内容时，可不必太在意这些细节，将学习重点放在对 LINQ 的整体把握方面。

1.1 什么是 LINQ

LINQ 是 Language Integrated Query 的缩写，翻译成中文就是语言集成查询。LINQ 是一系列的编程接口，借助于 LINQ 技术，可以使用一种统一的方式查询各种不同类型的数据。LINQ 是微软公司在 Visual Studio 2008 和.NET Framework 3.5 版本中一项突破性的创新，它在对象领域和数据领域之间架起了一座桥梁。LINQ 通过使用特定的语法，可以对数据库、对象以及 XML 等多种类型的数据进行查询操作。LINQ 既可在新项目中使用，也可在现有项目中与非 LINQ 查询一起使用，唯一的要求是项目应面向.NET Framework 3.5 版本。

下面是一段使用 LINQ 对数据库进行查询的代码。

```
//使用 LINQ 从数据库中查询数据并操作数据
var contacts =
from customer in db.Customers
where customer.Name.StartsWith("A") && customer.Orders.Count > 0
orderby customer.Name
select new { customer.Name, customer.Phone };
```



- from、where、orderby 和 select 都是 LINQ 的关键字。
- db、Customers 以及 Customer 是已定义实体类。
- 代码中运用了 C# 3.0 的一些新特性，如 var 等。
- 代码实现的逻辑是从数据库的 Customers 表中查询出满足要求的记录。

传统上，针对数据库的查询是将相关 SQL 语句包含在字符串中发送给数据库服务器来实现的，这种实现方式不能使用集成开发环境（如 Visual Studio 2008）提供的编译时类型检查功能和 IntelliSense 的支持。本例使用 LINQ 对数据库进行查询的代码中没有使用 SQL 字符串，使用的是强类型化的对象来实现数据库查询操作，这种实现方式可以使用集成开发环境提供的编译时类型检查功能和 IntelliSense 的支持，有助于开发者编写出更健壮、更安全的代码。

现实世界中存在着各式各样的数据源，为了查询这些数据源，开发者需要针对各种不同类型的数据源学习不同的查询语法，如关系数据库、XML 文档、Web 服务等不同类型的数据源就使用不同的查询语法。LINQ 使得查询成为.NET Framework 3.5 中的语言级组成部分，借助于 LINQ，可以使用一种统一的方式，利用语言关键字和熟悉的运算符针对各种不同类型的数据源编写查询。而且，在 Visual Studio 2008 中使用 LINQ 会获得编译时类型检查功能和 IntelliSense 的支持。如图 1-1 所示是在 Visual Studio 2008 中使用 C#语言编写 LINQ 查询时具有 IntelliSense 支持的效果。

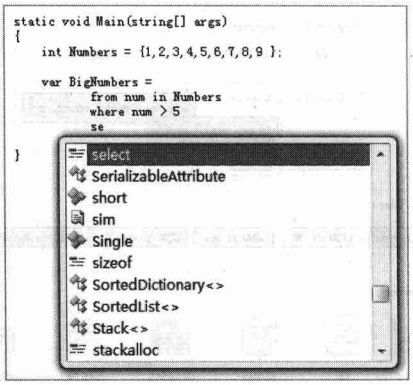


图 1-1　IntelliSense 支持的效果

1.2　LINQ 的设计目标

开发者可以使用各种各样的编程语言来编写程序，编程语言的选择可能取决于要解决问题的业务逻辑、个人的偏好、系统的部署平台以及所属公司的开发策略等。不管选用什么样的编程语言，与数据进行交互都是一项具有普遍意义的编程任务。数据可能是内存里面的一些对象集合、磁盘上的某个文件、数据库里面的多个数据表、一个 Excel 文件或者是来自 Web 的 xml 文档，在某些场合还可能需要处理包含一种或者几种类型的数据组合，每种类型的数据都有独有的访问模型。例如，当访问某种关系型数据库中的数据时，可以使用 SQL 语言；当访问 XML 数据时，可以使用 DOM（Document Object Model）编程接口或者 Xquery 等。总而言之，开发者需要使用不同的编程模型来访问不同的数据源。

在 LINQ 出现之前，曾有过一些试图将不同类型的数据访问接口进行统一的解决方案，如 ODBC，就可以使用一种统一的方式来访问一个 Excel 电子表格或者一个关系型数据库。但是，绝大多数此类的解决方案只能允许以相同的访问接口来访问关系模型数据，而对于一些层次模型数据，如一个对象数组或者一个 XML 文档，就无能为力了。LINQ 的出现就是试图在不同类型的数据之间，包括关系模型数据或者层次模型数据之间，提供一个统一的访问接口，开发者无需关心底层数据访问细节上的差异。

LINQ 采用一种开放性的设计架构，这种开放性不仅表现在其可以被多种.NET 语言所支持，还表现在通过为不同类型的数据源开发相应的 LINQ Provider，LINQ 可以在各种类型的数据源之间提供一个统一的访问接口。如图 1-2 所示为 LINQ 的设计模型图，从图中可以很清晰地看出，LINQ 通过使用一系列的 LINQ Provider，分别对 Object、XML、关系型数据库以及 DataSet 等不同类型数据源提供统一的访问接口，这意味着，开发者可以像操作内存对象一样来操作 XML 这类以前对开发者非常不友好（有过 XML 开发经历的开发者应该了解，使用 DOM 模型来操作 XML 非常繁琐，即便是 Xpath、Xquery 也不方便）的数据，相信读者在学习完本章后面的 3 个实例后，会有更深切的体会。

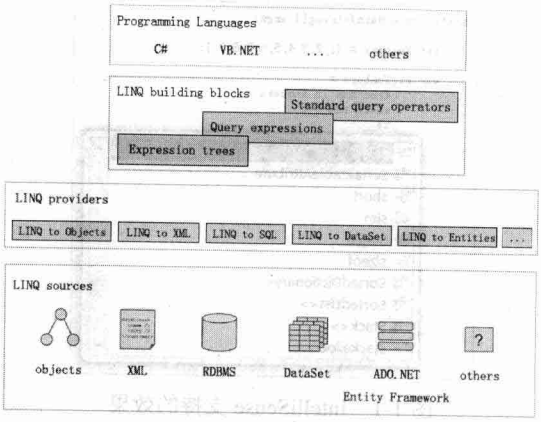


图 1-2　LINQ 的设计模型图

1.3　LINQ 的种类

LINQ 在不同类型的数据源之间提供了一个统一的访问接口，LINQ 作为.NET Framework 3.5 平台的组成部分，伴随着平台的发布，自身提供了 3 种 LINQ Provider，分别是 LINQ to OBJECT、LINQ to ADO.NET 和 LINQ to XML，如图 1-3 所示。

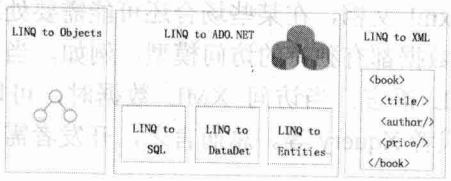


图 1-3　.NET Framework 3.5 平台的 LINQ 实现种类

1.3.1　LINQ to OBJECT

LINQ to OBJECT 是用来访问对象集合的编程接口，对象集合里面的元素之间可以具有层次结构。在.NET 中支持列举操作的各种对象类型，基本上都可以使用 LINQ to OBJECT 来进行操作。例如，可以使用 LINQ to OBJECT 来对一个整数类型数组中所有元素进行排序，也可以在一个自定义类型的集合中找出符合某些条件的元素子集合。

下面的代码实现了将一个整数类型数组 numbers 中所有元素按照从小到大的顺序排列，并放入集合 items 中的逻辑。

```
//将一个整数类型数组 numbers 中所有元素按照从小到大的顺序排列，并放入集合 items 中
using System;
using System.Collections.Generic;
using System.Text;
using System.Linq;
```

```
namespace Sample
{
    public class Test
    {
        static void Main(string[] args)
        {
            //使用集合初始化器初始化整数类型数组 numbers
            int[] numbers = { 10, 6, 8, 4, 9, 2, 1, 5, 0 };
            var items = from s in numbers
                        orderby s
                        select s;
            //输出结果
            foreach (var item in items)
                Console.WriteLine(item);
        }
    }
}
```

专家讲解

- 代码需要在支持 C# 3.0 的编译器中编译，如 Visual Studio 2008。
- 使用 LINQ to OBJECT 需要引用 System.Linq 命名空间。

1.3.2　LINQ to ADO.NET

LINQ to ADO.NET 是用来访问关系模型数据的编程接口，其可以进一步分为 LINQ to SQL、LINQ to Entities 和 LINQ to DataSet 这 3 个子类别，每个子类别针对特定的关系模型数据。其中，LINQ to SQL 在.NET 自定义类型（class）和数据库的物理表之间建立映射，通过操作自定义类型从而实现对数据库物理表的操作；LINQ to Entities 与 LINQ to SQL 有相似之处，但是 LINQ to Entities 并不是直接在数据库物理表和自定义类型之间建立映射，而是采用了一个概念上的实体数据模型，这项技术目前还在持续的研发中，本书将不会涉及到 LINQ to Entities 的内容；LINQ to DataSet 是使用 LINQ 来访问 DataSet 的接口。

代码演示

下面的代码实现了使用 LINQ to SQL 向数据库物理表中添加一条记录的逻辑。

```
//使用 LINQ to SQL 向数据库物理表中添加一条记录
Book book = new Book();                    //生成一个实体类对象
book.BOOKID = 99;
book.Book_Name = "C#程序设计";
book.Book_Author = "张三";
BookDataContext ctx = new BookDataContext();
                                           //只要与数据库通信，BookDataContext 对象必不可少
```

```
ctx.Book.InsertOnSubmit(book);          //将需要修改的对象添加到 BookDataContext 对象中
ctx.SubmitChanges();                    //通过 BookDataContext 对象将修改保存到数据库
```

专家讲解

上述代码是本章后面 LINQ to SQL 应用实例的一部分，请参考后面实例。

1.3.3 LINQ to XML

LINQ to XML 是用来访问 XML 数据的编程接口，XML 目前已经成为被各种开发平台所支持的一种数据标准。关于 XML，存在着各种各样的规格和模型，具体如下。

- XML Schema Definition（XSD）：用来定义 XML 文档的结构。
- Extensible Stylesheet Language for Transformations（XSLT）：将 XML 文档在不同的结构间转换。
- Document Object Model（DOM）：用来管理在内存中的 XML 文档。
- Simple Object Access Protocol（SOAP）：使用 XML 来实现平台间的互操作性。

各种规格和模型都有其独特的语法，这些语法使得操作 XML 对于开发者来讲并不轻松。LINQ to XML 提供了一个统一的编程接口，此编程接口与.NET Framework 高度集成，开发者使用 LINQ to XML 可以像操作.NET 集合对象一样来实现如创建、查找、查询以及管理 XML 内容等功能。

代码演示

下面的代码实现了使用 LINQ to XML 创建一个 XML 元素的逻辑。

```
//使用 LINQ to XML 创建一个 XML 元素
Books = new XElement("Books",
        new XElement("Book001",
        new XElement("Title", "C 程序设计"),
        new XElement("Author", "谭浩强")),
        new XElement("Book002",
        new XElement("Title", "算法与数据结构"),
        new XElement("Author", "张浩")));
```

专家讲解

- 代码是本章后面 LINQ to XML 应用实例的一部分，请参考后面实例。
- 使用 LINQ to XML 需要引用 System.Xml.Linq 命名空间。

上述代码所创建的 XML 内容如下。

```
<Books>
  <Book001>
    <Title>C 程序设计</Title>
    <Author>谭浩强</Author>
  </Book001>
```

```
    <Book002>
      <Title>算法与数据结构</Title>
      <Author>张浩</Author>
    </Book002>
</Books>
```

1.4　LINQ 应用实例

本节将通过 3 个分别使用 LINQ to OBJECT、LINQ to SQL 和 LINQ to XML 的实例，初次体验 LINQ 的强大功能以及使用 LINQ 的简单与高效。

1.4.1　LINQ to OBJECT 应用实例

【实例 1-1】将通过一个 C#控制台应用程序来展示如何使用 LINQ 操作对象集合。运行 Visual Studio 2008，新建项目，在打开的【新建项目】窗口的【项目类型】列表框中选择【Visual C#】，在【模板】列表框中选择【控制台应用程序】。注意，在【新建项目】窗口右上角的下拉列表框中选择【.NET Framework 3.5】平台类型，如图 1-4 所示。

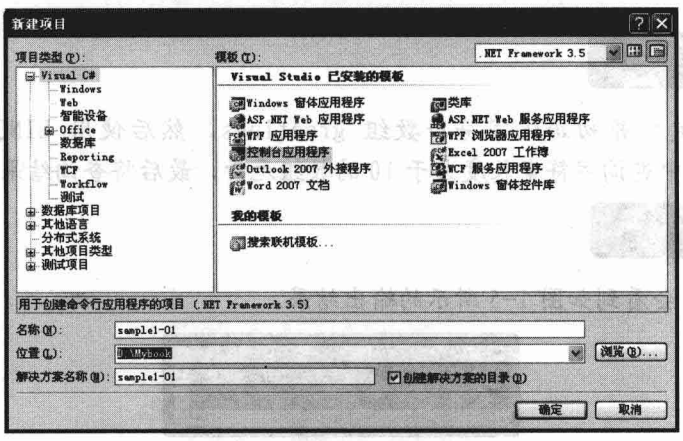


图 1-4　【新建项目】窗口

单击【确定】按钮后，在打开的 Program.cs 文件中输入下面的代码。

代码演示

```
//LINQ to OBJECT 实例代码
using System;
using System.Collections.Generic;
using System.Linq; //使用 Linq 需引入的命名空间
using System.Text;
namespace sample1_1
```