

微软.NET 程序设计系列

TP313.4
L13d

ADO.NET 程序设计

[美] Rebecca M. Riordan

著

李高健 孙瑛霖

译

Visual Studio .NET 产品组

审校

清华大学出版社

(京)新登字 158 号

内 容 简 介

.NET 框架的数据访问组件 ADO.NET 能与 XML 兼容平台上的任何一种组件协同工作。本书介绍了 ADO.NET 的对象模型以及如何使用这些模型开发数据绑定 Windows 窗体和 Web 窗体, ADO.NET 如何与 XML 交互以及如何通过 .NET 环境访问旧版本的 ADO。该书内容包括创建连接、数据命令和数据阅读器、使用数据集、数据表和数据视图、编辑和修改数据、Windows 窗体和 Web 窗体中的数据绑定、使用 XML 设计器修改数据结构、用 ADO.NET 读写 XML 以及通过 .NET 框架访问 ADO 等。本书旨在帮助初学者快速熟悉 ADO.NET 的组成和特性, 学会如何开发 Visual Basic .NET 和 Visual C# .NET 的数据库功能。该书浅显易懂, 示例丰富。

本书适合于有一定 Visual Basic 或 Visual C# 编程基础的 ADO 初学者, 对于想要学习和扩展数据库开发技术的开发者而言, 本书也是非常理想的学习工具。

Microsoft ADO.NET Step by Step

Microsoft Press

Copyright © 2002 by Microsoft Corporation

**Original English Language edition published by Microsoft Press, a Division of Microsoft Corporation
All rights reserved.**

No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher. For sale in the People's Republic of China only.

本书中文简体版由 Microsoft Press 授权清华大学出版社出版发行, 未经出版者书面许可, 不得以任何方式复制或抄袭本书的任何部分。

北京市版权局著作权合同登记号: 图字 01-2000-3535 号

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

书 名: ADO.NET 程序设计

作 者: [美] Rebecca M. Riordan

译 者: 李高健 孙瑛霖

出 版 者: 清华大学出版社(北京清华大学学研大厦, 邮编 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 蔡 颖 李祥青

印 刷 者: 北京牛山世兴印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×960 1/16 印张: 26.75 彩插页: 4 字数: 597 千字

版 次: 2002 年 7 月第 1 版 2002 年 7 月第 1 次印刷

书 号: ISBN 7-302-05663-3/TP·3340

印 数: 0001~5000

定 价: 48.00 元

目 录

前言	ix
----------	----

第 I 部分 ADO.NET 入门

第 1 章 ADO.NET 入门	3
1.1 关于对象的说明	3
1.2 ADO.NET 对象模型	4
1.2.1 数据提供程序	4
1.2.2 数据集	5
1.3 把数据绑定到简单的 Windows 窗体上	6
1.3.1 把连接和数据适配器添 加到窗体中	6
1.3.2 创建数据集	10
1.3.3 把控件简单绑定到 数据集上	12
1.3.4 把数据加载到数据集中	13
1.4 本章小结	14

第 II 部分 数据提供程序

第 2 章 创建连接	19
2.1 了解连接	19
2.2 创建连接	20
2.2.1 设计时连接	20
2.2.2 在运行时创建连接	23
2.3 使用 Connection 的属性	24
2.3.1 ConnectionString 属性	25
2.3.2 使用其他连接属性	27
2.4 Connection 方法	30

2.5 处理 Connection 事件	32
2.5.1 StateChange 事件	32
2.5.2 InfoMessage 事件	34
2.6 本章小结	34
第 3 章 数据命令和 DataReader	35
3.1 了解数据命令和 DataReader	35
3.2 创建数据命令	36
3.2.1 在设计时把数据命令 添加到窗体上	36
3.2.2 在运行时创建数据命令	37
3.3 命令属性	38
3.3.1 在设计时设置 Command 的属性	40
3.3.2 在运行时设置 Command 的属性	41
3.3.3 使用参数集合	42
3.4 Command 的方法	46
3.4.1 执行命令	47
3.4.2 DataReader	48
3.5 本章小结	54
第 4 章 数据适配器	56
4.1 了解数据适配器	56
4.2 创建数据适配器	57
4.2.1 使用【服务器资源管 理器】	57
4.2.2 使用【工具箱】	58
4.2.3 在运行时创建数据适配器	59
4.2.4 预览结果	60
4.3 数据适配器的属性	61
4.3.1 数据适配器命令	63

4.3.2 TableMappings 集合	64
4.4 数据适配器的方法	66
4.4.1 生成数据集和绑定数据	66
4.4.2 Fill 方法	68
4.4.3 Update 方法	72
4.5 处理数据适配器事件	74
4.5.1 OnRowUpdating 事件	74
4.5.2 OnRowUpdated 事件	79
4.6 本章小结	83
第 5 章 ADO.NET 中的事务处理	84
5.1 事务	84
5.2 创建事务	85
5.2.1 创建新事务	85
5.2.2 创建嵌套事务	90
5.3 使用事务	92
5.3.1 指派事务给命令	93
5.3.2 提交和回滚事务	93
5.4 本章小结	100

第Ⅲ部分

操作数据

第 6 章 数据集	103
6.1 了解数据集	103
6.2 创建数据集	104
6.2.1 创建类型化数据集	105
6.2.2 创建非类型化数据集	107
6.3 数据集属性	109
6.3.1 数据集的 Tables 集合	110
6.3.2 数据集的 Relations 集合	118
6.4 数据集方法	122
6.5 本章小结	126
第 7 章 数据表	128
7.1 了解数据表	128

7.2 创建数据表	129
7.2.1 创建单独的数据表	129
7.2.2 在数据集中创建数据表	133
7.3 数据表属性	135
7.3.1 使用数据表属性	136
7.3.2 Columns 集合	138
7.3.3 Rows 集合	141
7.3.4 Constraints 集合	146
7.4 数据表的方法	151
7.4.1 Select 方法	152
7.4.2 数据行的方法	154
7.5 数据表的事件	157
7.6 本章小结	157

第 8 章 数据视图

8.1 了解数据视图	159
8.2 创建数据视图	160
8.2.1 使用 Visual Studio	161
8.2.2 在运行时创建数据视图	162
8.3 数据视图的属性	165
8.3.1 数据列表达式	165
8.3.2 表达式排序	168
8.3.3 行状态过滤器	170
8.4 数据视图的方法	173
8.5 本章小结	175

第Ⅳ部分

使用 ADO.NET 对象

第 9 章 编辑和更新数据	179
9.1 编辑和更新数据	179
9.2 数据行的状态和版本	180
9.2.1 RowState 属性	180
9.2.2 行的版本	181
9.2.3 数据行的状态和版本	181
9.3 编辑数据集中的数据	183

9.3.1 添加数据行	183	11.1.2 使用 Parse 事件	238
9.3.2 删除数据行	185	11.2 在 Windows 控件中显示数据	240
9.3.3 改变数据行的值	187	11.2.1 简化数据输入	241
9.3.4 推迟对数据行值的修改	189	11.2.2 处理数据关系	246
9.4 更新数据源	192	11.2.3 使用数据窗格显示 分层数据	248
9.4.1 使用 DataAdapter 的 Update 方法	192	11.2.4 使用 TreeView 显示 分层数据	253
9.4.2 执行 Command 对象	195	11.2.5 查找数据	256
9.5 接受和拒绝数据集修改	200	11.3 在 Windows 窗体中验证数据	263
9.5.1 使用 AcceptChanges	200	11.3.1 数据变化事件	263
9.5.2 使用 RejectChanges	204	11.3.2 控制验证事件	267
9.6 本章小结	207	11.3.3 使用 ErrorProvider 组件	271
第 10 章 Windows 窗体中的 ADO.NET 数据绑定	208	11.4 本章小结	275
10.1 了解 Windows 窗体中的数 据绑定	208	第 12 章 Web 窗体中的数据绑定	277
10.2 把控件绑定到 ADO.NET 数据源上	210	12.1 了解 Web 窗体中的数据绑定	277
10.2.1 简单绑定控件属性	210	12.2 把控件绑定到 ADO.NET 数据源	279
10.2.2 复杂绑定控件属性	216	12.2.1 简单绑定控件属性	279
10.3 使用 BindingContext 对象	219	12.2.2 复杂绑定控件属性	283
10.4 使用 CurrencyManager 对象	220	12.2.3 使用 DataBinder 对象	286
10.4.1 CurrencyManager 的 属性	220	12.3 维护 ADO.NET 对象的状态	288
10.4.2 CurrencyManager 的 方法	225	12.3.1 在服务器上维护 ADO.NET 对象	288
10.4.3 CurrencyManager 的 事件	226	12.3.2 在页面上维护 ADO.NET 对象	290
10.4.4 使用 Binding 对象	228	12.4 在 Web 窗体中更新数据源	292
10.5 本章小结	231	12.5 本章小结	295
第 11 章 在 Windows 窗体中 使用 ADO.NET	232	第 13 章 在 Web 窗体中使用 ADO.NET	296
11.1 格式化数据	232	13.1 使用基于模板的 Web 控件	296
11.1.1 使用 Format 事件	233	13.1.1 使用数据网格控件	297
		13.1.2 使用数据列表控件	309

13.2 移动数据	315	第 15 章 读写 XML	360
13.3 本章小结	327	15.1 了解 ADO.NET 和 XML	360
第 V 部分		15.2 使用数据集的 XML 方法	361
ADO.NET 与 XML		15.2.1 GetXml 和 GetXmlSchema 方法	361
第 14 章 使用 XML 设计器	331	15.2.2 ReadXmlSchema 方法	367
14.1 了解 XML 架构	331	15.2.3 InferXmlSchema 方法	370
14.2 创建 XML 架构和类型化 数据集	332	15.2.4 ReadXml 方法	372
14.2.1 创建架构	333	15.2.5 WriteXmlSchema 方法	376
14.2.2 创建数据集	334	15.2.6 WriteXml 方法	378
14.2.3 了解架构属性	336	15.3 控制写 XML 的方式	381
14.3 在 XML 设计器中使用数据表	338	15.3.1 使用数据关系的嵌套 属性	382
14.3.1 把数据表添加到 XML 设计器中	339	15.3.2 使用数据列的 ColumnMapping 属性	384
14.3.2 创建键	342	15.4 XmlDataDocument 对象	387
14.3.3 创建关系	345	15.5 本章小结	390
14.4 使用元素	348	第 16 章 在 .NET 框架中使用 ADO	392
14.5 使用类型	351	16.1 了解 COM 互操作性	392
14.5.1 简单类型	351	16.2 在 .NET 框架中使用 ADO	394
14.5.2 复杂类型	353	16.2.1 建立一个对 ADO 的 引用	396
14.6 使用特性	356	16.2.2 创建 ADO 对象	398
14.6.1 特性的属性	356	16.3 在 .NET 框架中使用 ADOX	405
14.6.2 创建特性	357	16.4 本章小结	411
14.7 本章小结	358		

第 I 部分

ADO.NET 入门

ADO.NET 中最重要的部分就是它的对象模型，因此要开始学习 ADO.NET，首先就要对它的对象模型有所了解。

这一部分首先介绍 ADO.NET 的主要对象和它们之间的关系，怎样利用这些对象从数据存储的物理位置上获取数据，以及怎样保存数据。然后介绍了如何利用数据适配器配置向导创建一组对象，并将它们绑定到简单的数据窗体上。本部分的目的在于通过这些介绍和示范，让您对 ADO.NET 的基本概念有所了解。



ADO.NET 入门

本章学习目标

- 认识 ADO.NET 的主要对象以及它们之间的关系
- 利用数据适配器配置向导创建 Connection 和 DataAdapter 对象
- 自动生成数据集
- 将控件属性绑定到数据集
- 在运行时将数据载入数据集

与 .NET 框架的其他组件一样，ADO.NET 也包含了一组对象，这些对象进行交互以提供我们所需的功能。不过这样一来，我们学习这些对象模型的使用时会有一些麻烦，因为要了解任何一个对象，都必须了解它的全部细节。

为了解决这个问题，我们首先要建立一个概念上的框架。就是说，在学会任何一个对象的功能细节之前，要对每个对象的特性，以及对象之间的交互方式有一个大致的了解。

以下是本章中要学习的内容：首先介绍 ADO.NET 包含的主要对象，怎么利用这些对象从物理数据源获取数据，提供给用户，并从用户得到反馈。接着将建立一组对象，然后将它们绑定到简单的数据窗体上。

1.1 关于对象的说明

这一部分后面几章将依次介绍 ADO.NET 对象模型中的每个对象。实际上，因为对象之间的联系很紧密，我们不大可能孤立地、单独地看待其中任何一个对象。

当必须使用一种还未介绍过的方法或属性时，文中将会有所说明，提示这些方法或属性在哪一章中有详细介绍。

1.2 ADO.NET 对象模型

图 1.1 展示了 ADO.NET 对象模型中的主要对象。当然, 实际上的类库(class library)是很复杂的, 但我们会在后面处理这些难题。现在要了解的是 ADO.NET 对象模型中有哪些主要对象, 以及它们之间是如何交互的。

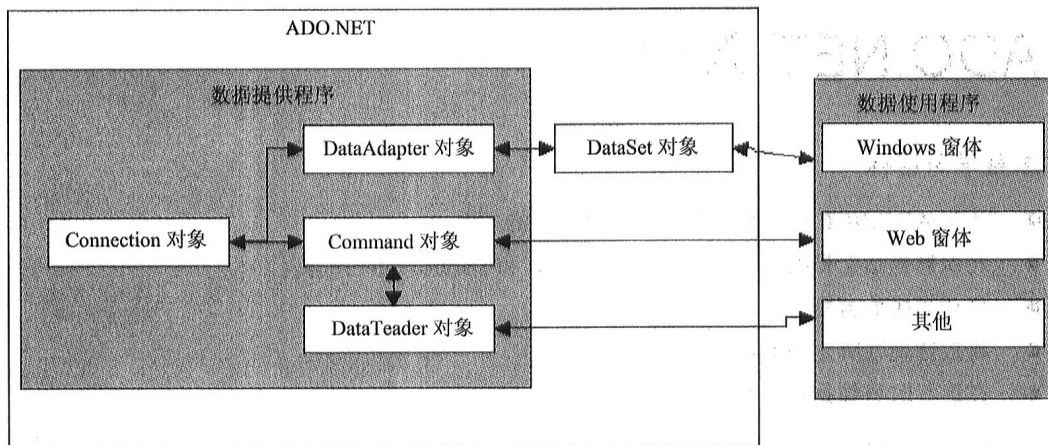


图 1.1

ADO.NET 的类由两个部分组成: 数据提供程序(Data Provider, 有时也叫托管提供程序)和数据集(DataSet)。数据提供程序负责与物理数据源的连接, 数据集代表实际的数据。这两个部分都可以和数据使用程序通信, 如 Web 窗体和 Windows 窗体。

1.2.1 数据提供程序

数据提供程序组件是数据源(data source)特有的。.NET 框架包含两种数据提供程序: 一种是 SQL Server 数据提供程序, 它已经为 SQL Server 7.0 或更高版本做了优化; 另一种是一般的数据提供程序, 它可以与任何一个 OLE DB 数据源通信。其他数据库(如 Oracle 和 DB2)的数据提供程序也是可用的, 另外我们也可以编写自己的数据提供程序(不过这已超出了本书范围)。

在.NET 框架下的两种数据提供程序具有功能相同的对象, 但这些对象的名称、部分属性或方法可能不同。SQL Server Provider 的对象名称以 SQL 为前缀(如 SqlConnection 对象); 而 OLE DB 对象名称以 OleDb 为前缀(如 OleDbConnection 对象)。

Connection 对象表示与一个数据源的物理连接。它的属性决定了数据提供程序(使用 OLE DB 数据提供程序时)、数据源、所连接到的数据库和连接期间用到的字符串。Connection 对象的方法比较简单: 打开和关闭连接, 以及改变数据库和管理事务。

Command 对象代表在数据源上执行的一条 SQL 语句或一个存储过程。对于一个 Connection 对象来说,可以独立地创建和执行不同的 Command 对象。DataAdapter(数据适配器)对象利用 Command 对象来处理后端数据集和数据源的通信。Command 对象支持 SQL 语句和存储过程,存储过程可返回单个值、一组或多组值,也可以不返回值。

DataReader 是一种快速、低开销的对象,用于从数据源中获取仅转发的、只读的数据流。DataReader 不能用代码直接创建,只能通过调用 Command 对象的 ExecuteReader 方法来创建。

在数据提供程序当中,DataAdapter 是功能最复杂的对象,它是 Connection 对象和数据集之间的桥梁。数据适配器包含 4 个命令对象: SelectCommand、UpdateCommand、InsertCommand 和 DeleteCommand。数据适配器用 SelectCommand 对象来填充数据集,其他 3 个对象在需要时用来改变数据源中的数据。

Microsoft ActiveX Data Object (ADO)

ADO.NET 的 Connection 对象和 Command 对象与它们在 ADO 中的功能大致相同(主要的区别在于缺少对服务器端游标的支持),而 DataReader 在 ADO 中的对应功能像一个流水游标。在 ADO 中没有与数据适配器和数据集对应的功能。

1.2.2 数据集

数据集(DataSet)是记录在内存中的数据,它的结构如图 1.2 所示。数据集有点像是一个简化的关系数据库,包含了表以及表与表之间的关系。要注意的是,数据集和数据源并没有连接在一起——数据集不知道自身所包含的数据来自哪里,实际上这些数据可能来自多个数据源。

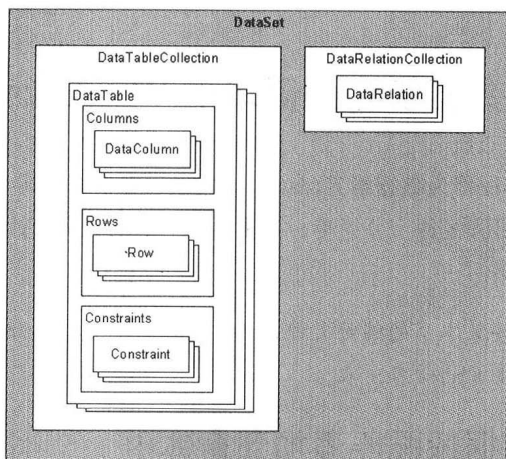


图 1.2

数据集由两个基本对象组成: DataTableCollection 和 DataRelationCollection。DataTableCollection 对象包含零个或多个 DataTable 对象,而 DataTable 对象又由 3 个集合组成: Column 集合、Row 集

合和 Constraint 集合。DataRelationCollection 对象包含零个或多个 DataRelation。

DataTable 的 Column 集合定义了组成数据表的列。除了 ColumnName 属性和 DataType 属性外,还可定义该列能否为空值(AllowDBNull),或定义其最大长度(MaxLength),或将它定义为可计算其值的表达式(Expression)。

数据表的 Rows 集合包含的是由 Column 集合定义的实际数据,但其中可能为空。对于每一行而言,数据表都会保留其原始值(original value)、当前值(current value)和建议值(proposed value)。这些功能大大简化了编程任务。

ADO

ADO.NET 的数据表提供了和 ADO 的 Recordset 对象相同的功能,但它们在对象模型中扮演着完全不同的角色。

数据表的 Constraints 集合包含零个或多个约束。在关系数据库中,约束用来维护数据的完整性。ADO.NET 支持两种形式的约束:外键约束(ForeignKey Constraint)和唯一键约束(UniqueKey Constraint)。外键约束维护关系的完整性(确保子表的行数据不会成为孤行);唯一键约束维护数据的完整性(确保表中不能有相同的行)。另外,数据表的 PrimaryKey 属性确保实体的完整性(它加强每一行数据的惟一性)。

数据集的 DataRelationCollection 对象包含了零个或多个数据关系。数据关系提供了一个简单的可编程界面,以实现从一个表中的主行导航到另一表中相关的行。举例来说,给出一个 Order,就可以通过数据关系轻松提取与该 Order 相关的数据行(注意,数据关系本身不能加强关系的完整性,而约束则可以)。

1.3 把数据绑定到简单的 Windows 窗体上

把数据连接到窗体的过程称为数据绑定(data binding)。数据绑定可以用代码实现,但是 Visual Studio .NET 设计器使这个过程变得十分简单。在本章中,我们将用设计器和向导快速简单地创建一个绑定了数据的 Windows 窗体。



要点

如果还没有安装本书的练习文件,请先完成“前言”部分的“安装和使用练习文件”,然后再回到本章的学习中来。

1.3.1 把连接和数据适配器添加到窗体中

绑定数据的第一步是创建数据提供程序的对象。Visual Studio 提供的【数据适配器配置向导】简化了这一过程。一旦添加了数据适配器,我们就可以使用 Visual Studio 中的【数据适配器预览】窗口检查它的配置是否正确。



提示 我们将在第2章详细介绍 Connection 对象, 在第4章详细介绍 DataAdapter(数据适配器)对象。

➤ 把连接添加到 Windows 窗体中

1. 从 Visual Studio 起始页打开 EmployeesForm 项目。
2. 在【解决方案资源管理器】中双击 Employees.vb (如果用 C# 则为 Employees.cs) 打开窗体。Visual Studio 在窗体设计器里显示窗体(图1.3)。

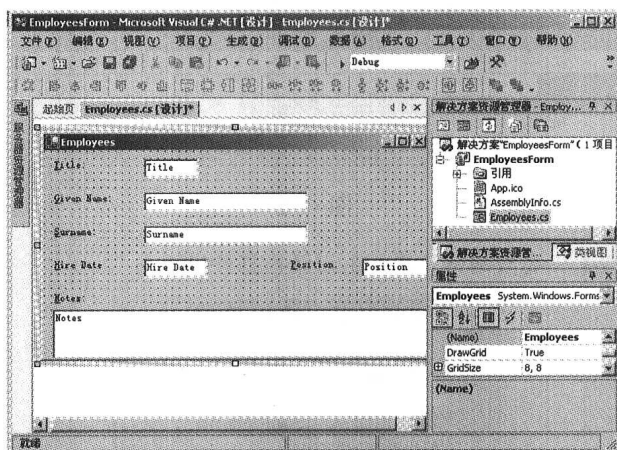


图 1.3

3. 从【工具箱】的【数据】选项卡上将一个 SQLDataAdapter 拖到窗体上。Visual Studio 将显示【数据适配器配置向导】的第一页(图1.4)。

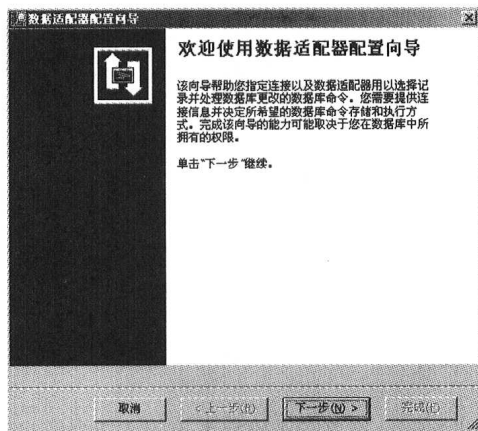


图 1.4

4. 单击【下一步】。【数据适配器配置向导】将显示一个页面, 要求用户选择一个连接(图1.5)。

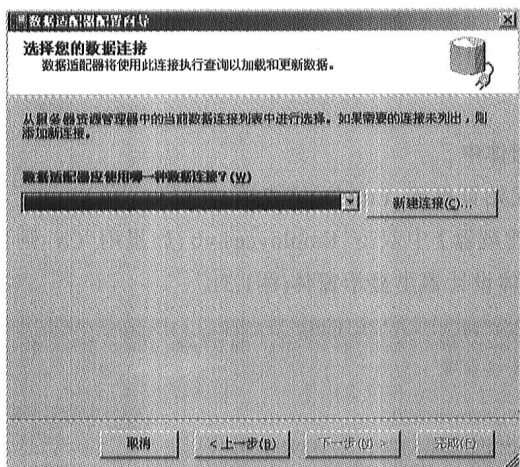


图 1.5

5. 单击【新建连接】。【数据链接属性】对话框打开(图1.6)。

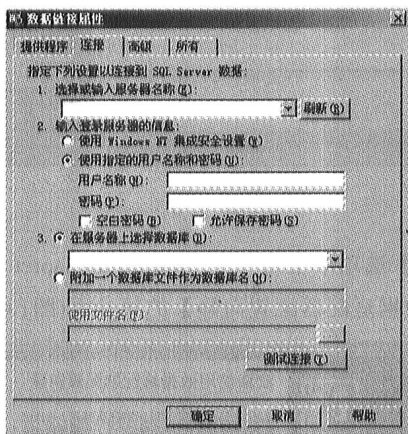


图 1.6

6. 指定服务器的名称和正确的登陆信息，选择 Northwind 数据库，然后单击【测试连接】按钮。【数据适配器配置向导】显示一个消息窗口，表明连接成功，如图1.7所示。



图 1.7



提示 如果不能确定如何完成第 6 步，请向系统管理员咨询。

7. 单击【确定】关闭消息窗口，再单击【确定】关闭【数据链接属性】对话框，然后单击【下一步】显示数据适配器配置向导的下一页(图1.8)。

【数据适配器配置向导】显示一个页面，请求选择一个查询类型。

8. 确认选中了【使用SQL语句】选项，然后单击【下一步】。

【数据适配器配置向导】弹出一个对话框，请求输入要用的 SQL 语句(图1.9)。

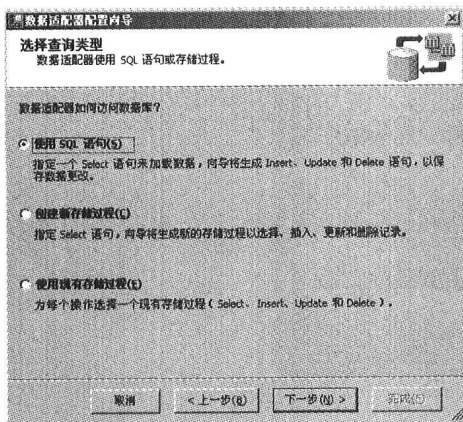


图 1.8

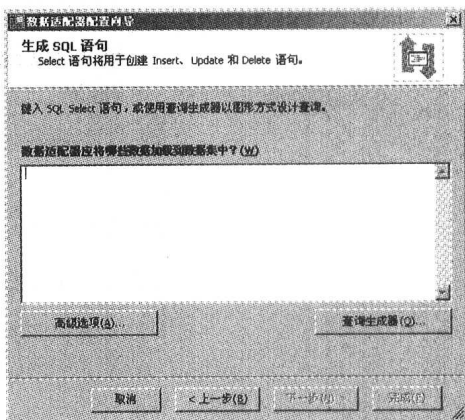


图 1.9

9. 单击【查询生成器】按钮。【数据适配器配置向导】打开【查询生成器】，显示【添加表】对话框(图1.10)。
10. 选择 Employees 表，单击【添加】，然后单击【关闭】(图1.11)。【添加表】对话框关闭时，Employees 表已被添加到【查询生成器】中。
11. 选择顶部窗格中以下字段的复选框：EmployeeID、LastName、FirstName、Title、TitleOfCourtesy、HireDate、Notes，将它们添加到查询中。【查询生成器】创建 SQL 命令(图1.12)。
12. 单击【确定】关闭【查询生成器】，然后单击【下一步】。

【数据适配器配置向导】弹出一个对话框，显示在窗体上添加Connection对象和数据适配器对象后的结果。

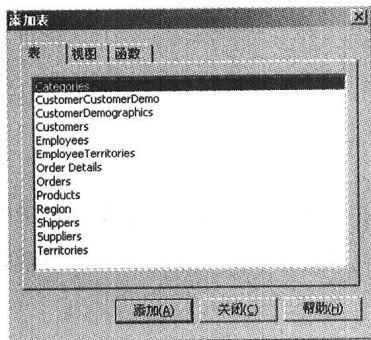


图 1.10

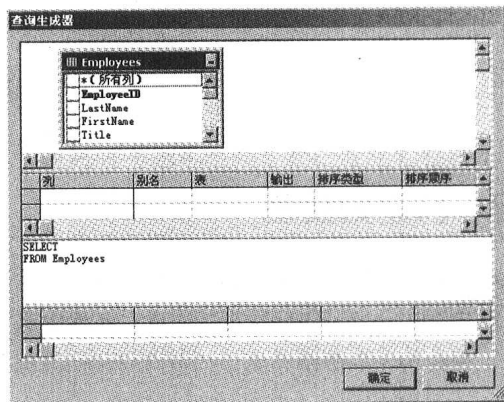


图 1.11



提示 我们将在第6章详细介绍数据集。

➤ **创建一个数据集**

1. 在【数据】菜单中，选择【生成数据集】。【生成数据集】对话框打开(图1.14)。

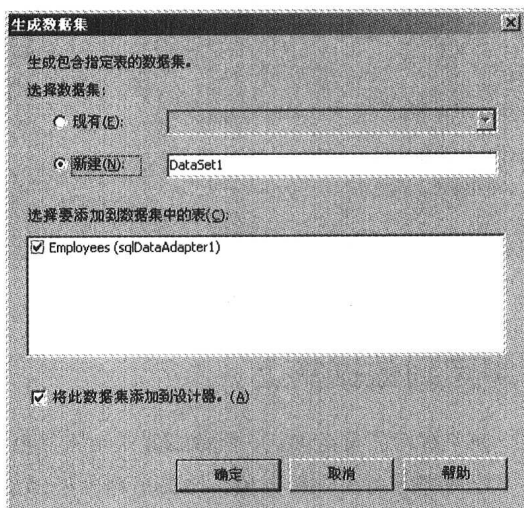


图 1.14

2. 在【新建】文本框中，输入dsEmployees(图1.15)。

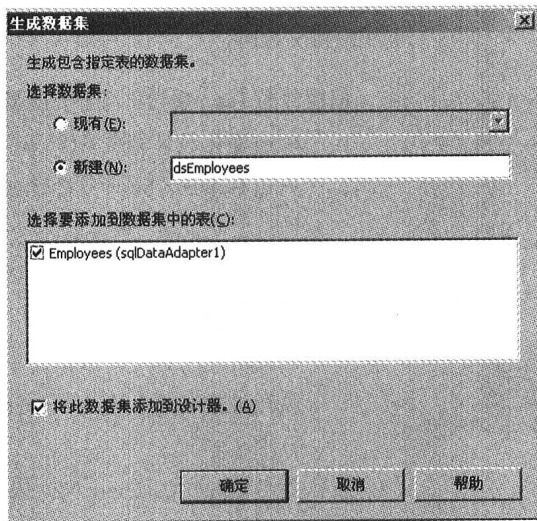


图 1.15

3. 单击【确定】。Visual Studio创建数据集的类，并且把类的一个实例添加到窗体设计器底部的窗格中(图1.16)。