

Visual  
Studio  
6.0应用系列

Visual  
6.0

Basic

中文版  
编程指南

京辉热点工作室 编著  
王克己 审校

民邮电出版社

Visual Studio 6.0 应用系列

# Visual Basic 6.0 中文版编程指南

京辉热点工作室 编著

王克己 审校

人民邮电出版社

## 图书在版编目 (CIP) 数据

Visual Basic 6.0 中文版编程指南/京辉热点工作室编著。—北京：人民邮电出版社，1999.9

(Visual Studio 6.0 应用系列)

ISBN 7-115-07963-3

I.V.... II.京... III.Basic 语言-程序设计 IV.TP312

中国版本图书馆 CIP 数据核字 (1999) 第 45953 号

Visual Studio 6.0 应用系列

### Visual Basic 6.0 中文版编程指南

---

- ◆ 编 著 京辉热点工作室
- 审 校 王克己
- 责任编辑 李 际
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号  
北京朝阳展望印刷厂印刷  
新华书店总店北京发行所经销
- ◆ 开本: 787 × 1092 1/16  
    印张: 24.25  
    字数: 606 千字  
    印数: 1 - 6 000 册
- 1999 年 9 月第 1 版  
    1999 年 9 月北京第 1 次印刷

ISBN 7-115-07963-3/TP·1211

---

定价: 38.00 元

本书重点介绍了 Visual Basic 6.0 中的许多新技术，如 ActiveX 数据对象、File System Objects 对象模型等等，用户通过使用这些新技术可以大大提高编程效率。本书介绍的另一个重点是 Visual Basic 的各种使用技巧，这些技巧大多散布于本书的各章中，而对于一些最常用的技巧，本书则专门用一章来详细介绍。

本书适合于有一定 Visual Basic 语言基础的用户和那些希望了解 Visual Basic 6.0 新技术的用户阅读。

## 前 言

在 Visual Basic 语言问世以前, 设计 Windows 应用程序是一件令人痛苦的事情。程序员必须充分了解 Windows 系统的各种知识, 记住数以百计的各种 Windows 消息。而在 Visual Basic 出现以后, 一切似乎变得简单多了。Visual Basic 把 Windows 的各种细节隐藏起来, 程序员需要做的只是在表单中“画”出需要的控件, 然后为这些控件的事件编写代码。与此同时, Visual Basic 提供了强大的功能, 使得它足以应付用户的绝大部分要求。正是由于 Visual Basic 功能强大、使用简单, 使得它在短短的数年间从 16 位发展到 32 位, 从 Visual Basic 1.0 发展到 Visual Basic 6.0。作为 Visual Basic 的最新版本, Visual Basic 6.0 提供了更新、更强大的功能。

本书作为中文版 Visual Basic 6.0 的编程指南, 它并不是一本 Visual Basic 语言的入门书籍, 它主要是针对两种读者: 一是那些有了一定 Visual Basic 语言基础而希望提高自己编程水平的读者; 二是那些对 Visual Basic 以前版本有一定了解而希望掌握 Visual Basic 6.0 中新技术的读者。正是由于这个原因, 本书的重点是 Visual Basic 6.0 中的新技术和 Visual Basic 的使用技巧。本书中第一、二、三、五章的全部和第四、六章的一部分都是在介绍 Visual Basic 6.0 中的新技术, 如 ActiveX 数据对象、File System Objects 对象模型等。而在介绍这些新技术的同时, 也介绍了 Visual Basic 中的一些使用技巧, 在第六章中则集中介绍一些使用技巧。

最后需要说明的一点是, 本书中的大部分例子只是用于演示功能, 因此它们没有错误处理部分或只有一个简单的错误处理。用户在实际运用过程中, 必须自己编写一个完善的错误处理过程。

本书由京辉热点工作室编写, 张严辞执笔, 参加本书的策划、编写等工作的还有王克己、王永辉、李涛、张雁芳、曹伟、张锁晋、武和平、齐雅昭、孙培兴、赵蒙、周德明、罗中雁、付惠敏、钱越等, 全书由王克己审校。由于编者水平有限, 书中错漏之处难免, 恳请读者批评指正。

编者  
1999.7

<b>第一章 ActiveX 数据对象基础</b> .....	1
1.1 ActiveX 数据对象概述.....	1
1.1.1 ActiveX 数据对象的优点.....	1
1.1.2 数据访问的步骤.....	2
1.1.3 ActiveX 数据对象的对象模型.....	2
1.2 数据源和数据库.....	2
1.2.1 关于数据库的说明.....	3
1.2.2 为系统添加一个数据源.....	4
1.3 建立 ActiveX 数据对象与数据源的连接.....	6
1.4 建立并执行一个命令.....	7
1.5 浏览 RecordSet 对象中的记录.....	8
1.6 在 RecordSet 对象中插入、删除、修改一条记录.....	12
1.6.1 插入一条记录.....	12
1.6.2 删除当前行.....	13
1.6.3 修改当前行.....	13
1.6.4 将修改后的结果返回给数据源.....	13
1.7 本章回顾.....	20
<b>第二章 ActiveX 数据对象深入编程</b> .....	23
2.1 与数据源建立连接的多种方法.....	23
2.1.1 指定 Connection 对象的 ConnectionString 属性.....	23
2.1.2 使用 Recordset 对象的 Open 方法.....	25
2.2 打开 Recordset 对象的多种方法.....	26
2.2.1 Connection 对象的 Execute 方法.....	26
2.2.2 Command 对象的 Execute 方法.....	27
2.2.3 Recordset 对象的 Open 方法.....	27
2.2.4 3 种方法的比较.....	28
2.3 在 Recordset 对象中使用书签.....	29
2.4 快速浏览 Recordset 对象中的数据.....	32
2.5 过滤 Recordset 对象中的数据.....	33
2.6 查找 Recordset 对象中的数据.....	40
2.7 Recordset 对象的游标类型.....	46
2.7.1 Static.....	47
2.7.2 Dynamic.....	47
2.7.3 Forward-Only.....	47

2.7.4	Keyset.....	47
2.7.5	选择游标的位置.....	47
2.8	Recordset 对象的锁定类型.....	56
2.9	Recordset 对象的更新模式.....	56
2.9.1	立即模式.....	57
2.9.2	批模式.....	57
2.10	事务处理.....	67
2.11	参数化查询.....	76
2.11.1	生成一个 Parameter 对象.....	77
2.11.2	设置 Parameter 对象的属性.....	78
2.11.3	把 Parameter 对象加入到 Command 对象.....	78
2.11.4	进行参数化查询.....	78
2.12	控制 Recordset 对象的行数.....	83
2.13	使 ForwardOnly 游标具有向后卷动的功能.....	84
2.14	执行异步查询.....	89
2.15	更新 Recordset 对象中的数据.....	90
2.16	检查 Recordset 对象是否支持某项功能.....	94
2.17	创建 Recordset 对象的快捷方法.....	101
2.18	在一个 Command 对象中设置多个 SQL 语句.....	101
2.19	本章回顾.....	107
<b>第三章</b>	<b>数据环境设计器及 ADO 控件.....</b>	<b>109</b>
3.1	数据环境设计器的基本使用方法.....	109
3.2	关系继承.....	114
3.3	分组.....	118
3.4	对 Command 对象中的字段进行计算.....	121
3.5	在控件中使用 DataEnvironment 对象.....	126
3.6	与 ADO 有关的控件.....	127
3.6.1	ADO 数据控件.....	127
3.6.2	DataList 控件和 DataCombo 控件.....	130
3.6.3	DataGrid 控件.....	135
3.6.4	Hierarchical FlexGrid 控件.....	147
3.7	本章回顾.....	152
<b>第四章</b>	<b>面向对象编程.....</b>	<b>155</b>
4.1	实现一个基本的类模块.....	155
4.2	类模块中的属性.....	168
4.2.1	对属性值的有效性进行检查.....	169
4.2.2	定义只读属性.....	170
4.2.3	Property Set 语句.....	170
4.2.4	注意 Variant 类型的参数.....	171

4.2.5 将数组作为属性 .....	171
4.3 类模块中的事件 .....	172
4.4 Visual Basic 中的常用对象 .....	176
4.4.1 Dictionary 对象 .....	176
4.4.2 Err 对象 .....	179
4.4.3 Screen 对象 .....	179
4.4.4 App 对象 .....	180
4.5 本章回顾 .....	180
<b>第五章 File System Objects 对象模型 .....</b>	<b>183</b>
5.1 FSO 概述 .....	183
5.2 FileSystemObject 对象 .....	184
5.2.1 创建一个 FileSystemObject 对象 .....	184
5.2.2 显示系统中所有驱动器的类型和名称 .....	184
5.2.3 判断一个驱动器是否存在 .....	186
5.2.4 判断一个文件夹是否存在 .....	189
5.2.5 拷贝、移动、删除、建立一个文件夹 .....	190
5.2.6 获取父文件夹及系统中的特定文件夹 .....	195
5.2.7 建立一个路径 .....	198
5.2.8 判断一个文件是否存在 .....	198
5.2.9 拷贝、移动、删除、新建一个文件 .....	201
5.2.10 生成一个临时文件名 .....	205
5.3 Drive 对象 .....	206
5.3.1 Drive 对象的属性 .....	206
5.3.2 显示系统中所有驱动器的信息 .....	207
5.4 Folder 对象 .....	209
5.4.1 Folder 对象的属性 .....	209
5.4.2 Folder 对象的方法 .....	209
5.4.3 显示驱动器中所有文件夹的信息 .....	210
5.5 File 对象 .....	215
5.5.1 File 对象的属性 .....	215
5.5.2 File 对象的方法 .....	215
5.5.3 查找文件 .....	216
5.6 TextStream 对象 .....	222
5.6.1 TextStream 对象的属性 .....	222
5.6.2 TextStream 对象的方法 .....	222
5.6.3 建立一个新文件 .....	222
5.6.4 读取一个文件 .....	223
5.7 本章回顾 .....	224
<b>第六章 Visual Basic 6.0 常用技巧 .....</b>	<b>227</b>



6.1	Visual Basic 集成环境的使用技巧 .....	227
6.1.1	如何注释一段代码 .....	227
6.1.2	如何在多个函数间快速切换 .....	228
6.1.3	如何建立自己的工程模板 .....	229
6.2	Visual Basic 控件使用技巧 .....	229
6.2.1	如何自动选中文本框中的所有内容 .....	229
6.2.2	如何弹出文本框自己的弹出式菜单 .....	229
6.2.3	如何避免按回车键时喇叭发出声音 .....	230
6.2.4	如何限制用户在文本框中的输入 .....	230
6.2.5	如何快速对一个字符串数组进行排序 .....	231
6.2.6	如何动态载入控件 .....	232
6.2.7	ListBox 控件的 ListIndex 属性和 Click 事件 .....	232
6.2.8	转换用户的输入 .....	233
6.2.9	如何滚动显示一张大图片 .....	235
6.2.10	如何改善日期时间的输入界面 .....	238
6.2.11	如何在 ComboBox 控件中加入图片 .....	239
6.3	Visual Basic 语言使用技巧 .....	242
6.3.1	如何自定义一个数据类型 .....	242
6.3.2	如何快速给一个数组赋值 .....	243
6.3.3	如何在保持原有数据的基础上重新定义动态数组的大小 .....	244
6.3.4	如何将数组作为函数的返回类型 .....	244
6.3.5	如何快速搜索一个一维字符串数组 .....	245
6.3.6	如何按不同格式显示时间和日期 .....	246
6.3.7	如何格式化输出数字 .....	247
6.3.8	如何从右到左搜索一个字符串 .....	247
6.3.9	如何将一个数组中的所有元素连接成一个字符串 .....	248
6.3.10	如何将一个字符串分拆成一个数组 .....	249
6.3.11	如何实现字符串替代 .....	249
6.3.12	如何颠倒字符串中的所有字符 .....	250
6.3.13	CallByName 函数 .....	250
6.3.14	如何将一个过程或函数的参数定义为可选参数 .....	252
6.3.15	如何将参数传递给表单 .....	252
6.4	本章回顾 .....	253
第七章	使用其他应用程序 .....	255
7.1	SendKeys 语句 .....	255
7.2	动态数据链接 (DDE) .....	259
7.2.1	DDE 概述 .....	259
7.2.2	建立一个典型的链接 .....	260
7.2.3	初始化链接 .....	261

7.2.4	执行命令.....	270
7.2.5	客户程序向服务器程序提供数据.....	273
7.2.6	服务器程序向客户程序提供图形数据.....	274
7.2.7	实例.....	277
7.3	使用其他程序中的对象.....	279
7.3.1	概述.....	279
7.3.2	建立对外部应用程序中对象的引用.....	280
7.3.3	Microsoft Word 所提供的对象.....	283
7.3.4	对一段文字进行统计.....	288
7.3.5	将一个 Grid 输出到 Word 程序.....	291
7.3.6	拼写检查.....	295
7.4	本章回顾.....	308
<b>第八章</b>	<b>通用查询模块.....</b>	<b>309</b>
8.1	通用查询模块概述.....	309
8.2	设计思路.....	310
8.3	定义一个查询类.....	310
8.3.1	定义数据类型.....	310
8.3.2	定义类的属性.....	312
8.3.3	定义类的方法.....	312
8.4	创建一个 UserControl.....	317
8.4.1	实现功能.....	317
8.4.2	UserControl 的布局.....	317
8.4.3	UserControl 的属性.....	318
8.4.4	UserControl 的方法.....	321
8.4.5	UserControl 的事件.....	323
8.4.6	限制用户在 TextBox 控件中的输入.....	324
8.4.7	转化用户在 TextBox 控件中的输入.....	324
8.4.8	填充 ListBox.....	326
8.4.9	实现 ListBox 的多选和单选.....	328
8.4.10	UserControl 中的错误处理.....	328
8.4.11	UserControl 的完整代码.....	332
8.5	设计查询界面.....	342
8.5.1	查询界面的控件布局.....	342
8.5.2	表单中的 Public 变量.....	344
8.5.3	表单的 Load 事件.....	345
8.5.4	comboFields 的 Click 事件.....	346
8.5.5	comboOperator 的 Click 事件.....	349
8.5.6	增加一个查询条件.....	350
8.5.7	修改一个查询条件.....	353

8.5.8	生成 SQL 语句 .....	354
8.5.9	查询界面完整的代码 .....	358
8.6	测试通用查询模块 .....	372
8.6.1	设置查询所涉及到的字段 .....	372
8.6.2	设置查询所涉及到的字段的有效值 .....	373
8.6.3	设置结果集所涉及到的字段 .....	374
8.6.4	设置查询所涉及到的表及表之间的关系和连接字符串信息 .....	374
8.6.5	进行查询 .....	374
8.6.6	具体例子 .....	375
8.7	本章回顾 .....	378

## ActiveX 数据对象基础

ActiveX 数据对象 (ActiveX Data Object) 是 Visual Basic 6.0 提供的一种新的数据访问 (Data Access) 方法。在本章中将首先介绍 ActiveX 数据对象的一些基本知识, 如 ActiveX 数据对象的优点、对象模型等等, 然后介绍 ActiveX 数据对象的基本使用方法, 如怎样通过 ActiveX 数据对象与数据源建立连接, 怎样执行查询以及如何对取回的数据进行浏览、插入、删除等等。通过本章的学习, 希望用户能对 ActiveX 数据对象有一个基本的认识。

### 1.1 ActiveX 数据对象概述

ActiveX 数据对象与 Data Access Object (DAO)、Remote Data Object (RDO) 一样, 都是一种数据访问的方法。但与以前 Visual Basic 提供的数据库访问方法相比, ActiveX 数据对象的速度更快, 使用更方便, 内存占用更少。在本节中将介绍 ActiveX 数据对象的一些基本知识, 如 ActiveX 数据对象的优点, 用 ActiveX 数据对象进行数据库访问的基本步骤, ActiveX 数据对象的对象模型等等。

#### 1.1.1 ActiveX 数据对象的优点

具体说来, ActiveX 数据对象有如下的一些优点:

- 大部分的 ActiveX 数据对象是互相独立的, 用户可以根据自己的需要来创建 ActiveX 数据对象。这种各个对象之间的相互独立性是 ActiveX 数据对象与以前的一些数据库访问方法 (如 RDO, ADO) 一个很大的不同, 它使得访问数据库的手段更加灵活有效。
- ActiveX 数据对象提供的批更新模式使得用户能够在一次数据库更新中更新多条记录。这可以减轻网络和服务器的负担。
- ActiveX 对象支持多种游标类型。
- 用户可以决定返回的记录条数。通过这个特性, 用户可以实现一些用 SQL 语句较难实现的功能。
- 用户可以在一次查询中返回多个结果集。
- 完善的错误处理。

### 1.1.2 数据访问的步骤

用 ActiveX 数据对象进行数据访问的基本步骤如下：

(1) 与数据源建立连接。

(2) 设置一个命令来访问数据源的数据。

(3) 执行这个命令。

(4) 如果命令的执行导致返回数据库中表的一些行，则把这些行存放在一个缓冲中以便于对其进行操作。

(5) 对缓冲中的数据进行修改。

(6) 修改之后，把修改后的结果返回给数据源。

需要指出的是以上步骤只是用 ActiveX 数据对象进行数据访问的基本步骤。由于 ActiveX 数据对象中的各个对象是各自独立的，因此用户可以在以后的章节中看到，在实际运用中，有些步骤将被省略。

### 1.1.3 ActiveX 数据对象的对象模型

ActiveX 数据对象中的对象有：Connection、Command、Parameter、Recordset、Field、Error。它们之间的关系如图 1-1 所示。

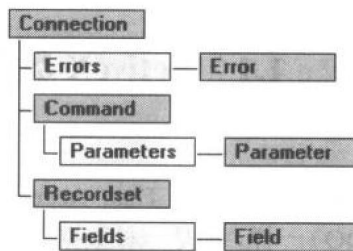


图 1-1 ActiveX 数据对象的对象模型

其中最主要的对象是 Connection、Command 和 Recordset 这 3 个对象。Connection 对象主要负责与数据源建立连接；Command 对象主要用于设置访问数据源所需的命令；Recordset 对象主要用来存放从结果集中取回的记录。而在这 3 个对象中，最重要的又是 Recordset 对象，用户可以在以后的章节中看到，在实际运用中，可以不创建 Connection 对象和 Command 对象而直接创建一个 Recordset 对象。这种各个对象之间的相互独立性是 ActiveX 数据对象与以前的一些数据访问方法（如 RDO，ADO）一个很大的不同，它使得访问数据的手段更加灵活有效。

除了这 3 个对象以外，Error 对象主要用来存放访问数据源时所发生错误的信息，这个对象在用户的错误处理代码中是必不可少的。Parameter 对象主要用来进行参数化查询。而 Field 对象对应查询返回的各个字段。

## 1.2 数据源和数据库

在开始叙述 ActiveX 数据对象基本编程之前，将首先对本书所用的数据源和数据库的有

关内容作一个说明。

### 1.2.1 关于数据库的说明

本书使用的数据库是建立在 Microsoft SQL Server 上的一个名为“test”的数据库，它包含了 6 个表：Product、SalesOrder、SalesOrderDetail、Supplier、Employee、Region。表的具体结构说明如下。

#### 一、表 Product 的结构

表 Product 中保存了有关产品的一些信息，表的结构如表 1-1 所示。

**表 1-1 表 Product 的结构**

字段名	字段描述	字段类型
ProductName	产品名称	Char (30)
ProductID	产品编号	Char (10)
ProductType	产品类型	Char (30)
ProductUnit	单位	Char (10)
ProductBrand	产品品牌	Char (30)
Price	产品价格	Float

主键：ProductID

#### 二、表 SalesOrder 的结构

表 SalesOrder 中保存了订单的一些信息，其结构如表 1-2 所示。

**表 1-2 表 SalesOrder 的结构**

字段名	字段描述	字段类型
OrderID	订单编号	Char (10)
OrderType	订单类型	Int
SupplierID	供货商编号	Char (10)
MakeDate	制订日期	Date
Maker	制订人	Char (10)

主键：OrderID

外键：Supplier.SupplierID, SalesOrder.Maker

#### 三、表 SalesOrderDetail 的结构

表 SalesOrderDetail 中保存了每张订单的明细信息，其结构如表 1-3 所示。

**表 1-3 表 SalesOrderDetail 的结构**

字段名	字段描述	字段类型
OrderID	订单编号	Char (10)
ProductID	货品编号	Char (10)
OrderCount	订货数量	Int

主键：OrderID + ProductID

外键：SalesOrder.OrderID, Product.ProductID

#### 四、表 Supplier 的结构

表 Supplier 中保存了有关供货商的一些信息，其结构如表 1-4 所示。

**表 1-4 表 Supplier 的结构**

字段名	字段描述	字段类型
SupplierID	供货商编号	Char (10)
SupplierName	供货商名称	Char (50)
Header	负责人	Char (10)
Address	供货商地址	Char (100)
Telephone	供货商电话	Char (20)

主键: SupplierID

#### 五、表 Employee 的结构

表 Employee 中保存了有关雇员的一些信息，其结构如表 1-5 所示。

**表 1-5 表 Employee 的结构**

字段名称	字段描述	字段类型
EmployeeID	雇员编号	Char (10)
EmployeeName	雇员姓名	Char (10)
RegionID	雇员所在区域编号	Char (10)

主键: EmployeeID

外键: Region.RegionID

#### 六、表 Region 的结构

表 Region 中保存了有关区域的一些信息，其结构如表 1-6 所示。

**表 1-6 表 Region 的结构**

字段名称	字段描述	字段类型
RegionID	区域编号	Char (10)
RegionName	区域名称	Char (50)

主键: RegionID

#### 七、表之间的连接关系

这 6 个表之间的连接关系如图 1-2 所示。

#### 1.2.2 为系统添加一个数据源

若没有特殊说明，本书所用到的数据源名都假设为“test”，用户 ID 为空，用户的 Password 为空，使用的数据库就是 1.2.1 节中的数据库 test。在使用本书的例子之前，用户必须在 Windows 系统下添加一个名为“test”的数据源。添加一个数据源的步骤如下：

- (1) 进入 Windows 中的“控制面板”。
- (2) 在“控制面板”中双击“ODBC”。
- (3) 在“User DSN”选项卡中单击“Add”按钮。
- (4) 在“Create New Data Source”中选择“SQL Server”，然后单击“完成”。如图 1-3

所示。

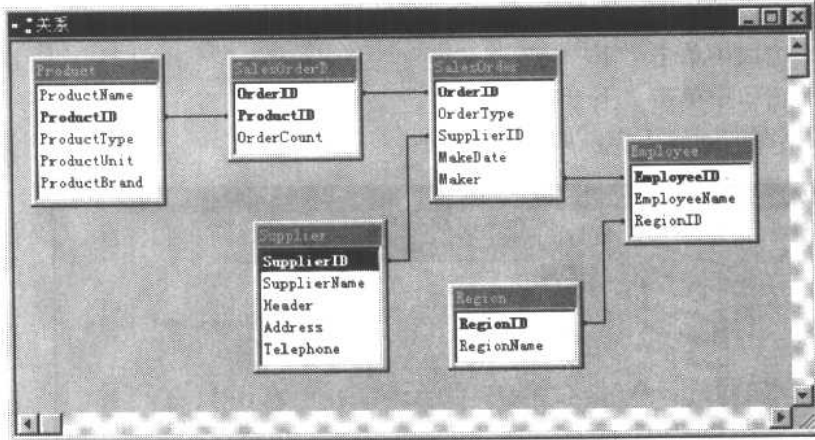


图 1-2 表之间的连接关系

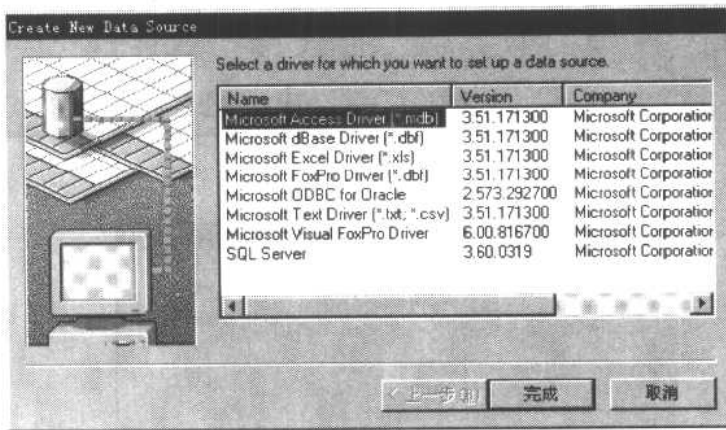


图 1-3 新建一个数据源

(5) 在下一个窗口中按图 1-4 所示填写有关数据源名称、描述以及所连接的 SQL Server 等信息，然后单击“下一步”。

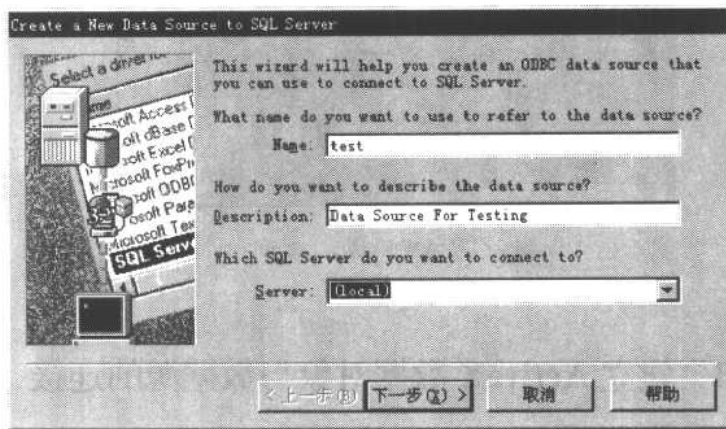


图 1-4 设置数据源的名称以及所连接的 SQL Server



- (6) 在下一个窗口中单击“下一步”。
- (7) 在下一个窗口中按图 1-5 所示填写所用的数据库名称等信息，然后单击“下一步”。
- (8) 在下一个窗口中单击“下一步”。
- (9) 在下一个窗口中单击“下一步”。
- (10) 完成数据源的新建工作，如图 1-6 所示。

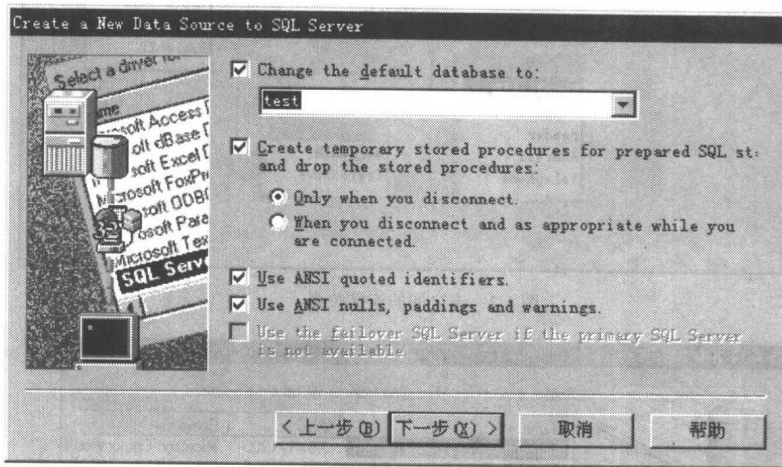


图 1-5 选择数据库

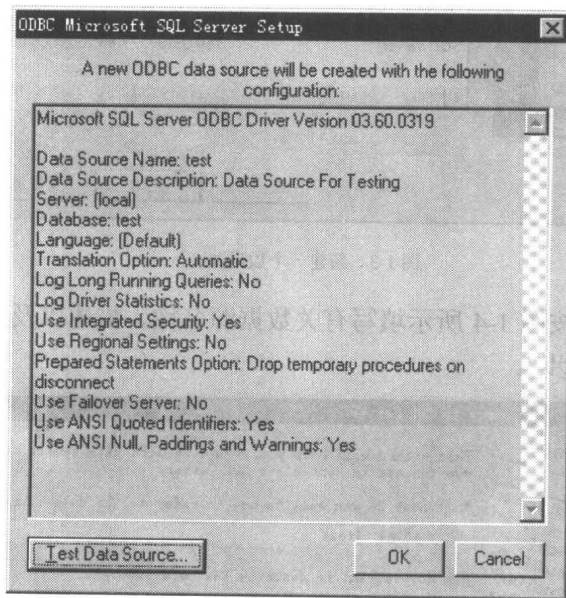


图 1-6 完成数据源的新建工作

### 1.3 建立 ActiveX 数据对象与数据源的连接

ActiveX 数据对象与数据源之间连接的建立主要是通过 Connection 对象来实现的。而在