

SAMS

北京科海培训中心

6.0

Visual Basic

数据库开发

[美] E.Winemiller, J.Roff
B.Heyman, R.Groom 著
顾 斌 杨德斌 译



清华大学出版社

Visual Basic 6.0 数据库开发

[美] E. Winemiller, J. Roff

B. Heyman, R. Groom 著

顾斌 杨德斌 译

清华大学出版社

(京)新登字 158 号

著作权合同登记号:01-98-2858

内 容 提 要

本书译自美国 How-To 畅销系列丛书之一《Visual Basic 6.0 Database How-To》。

全书以 120 个 Visual Basic 6.0 数据库问题的解决方案和大量的实例讲解了 VB 数据库开发中的各种问题。内容包括:创建记录集、参数查询以及设计和实现自己的 Visual Basic 数据库;使用数据访问对象(DAO)访问数据库;使用 Microsoft Data Report 和 Crystal Report;把你的数据库和数据连到 Web 站点或 Intranet;开发一个多用户访问的安全数据库;使用 ActiveX 自动化任务以及来自专家的高级数据库技术等内容。

本书内容丰富,编排独到,针对用 VB6 每一开发细节,以提问-分析-解决-总结-评论的方式向读者一一介绍。本书适用于各层次的 VB 程序应用与开发人员,也可供大学计算机专业的学生和教师作为参考书。

Visual Basic 6.0 Database How-To

Copyright © 1998 by Sams Publishing

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher.

本书中文简体字版由美国麦克米兰出版集团授权科海培训中心和清华大学出版社。未经出版者书面允许,不得以任何方式复制或抄袭本书内容。

版权所有,盗版必究。

本书封面贴有 PRENTICE HALL 激光防伪标志,无标志者不得进入各书店。

书 名: Visual Basic 6.0 数据库开发

原著者: E. Winemiller, J. Roff, B. Heyman, R. Groom

译 者: 顾斌 杨德斌

出版者: 清华大学出版社(北京清华大学校内,邮编 100084)

印刷者: 北京门头沟胶印厂

发 行: 新华书店总店北京科技发行所

开 本: 850×1168 毫米 印张: 47.25 字数: 1150 千字

版 次: 1999 年 9 月第 1 版 2000 年 7 月第 3 次印刷

印 数: 10001~13000

书 号: ISBN 7-900622-12-8

定 价: 85.00 元

前 言

关于本书

自从 Visual Basic 3.0 发布以来, Visual Basic 便成为广大程序员进行数据库编程的一种选择工具。Microsoft 在 Visual Basic 3.0 中首次引入了 DAO, 随后在 4 版中又推出了 RDO, 再接下来在 5 版中又加入了创建“健壮”的 ActiveX 组件的能力。随着版本的不断升级, Visual Basic 的功能越来越强大, 从而使数据库编程也越来越容易。

Visual Basic 强大的数据库功能随着第 6 版的推出再一次得到增强, 新的工具和方法如 ADO, OLE-DB 以及 Microsoft Data Reporter 成为引人瞩目的焦点, 这些新的东西意味着什么呢? 它们能为你做些什么? 更重要的是你怎样才能快速地掌握它们呢?

针对以上各种问题, 我们出版了《Visual Basic 6.0 数据库开发》一书。这本书对每一种主要的数据访问方法都进行了深入的探讨, 并给出了大量的实例。与所有其他受到读者喜爱的 How-To 风格系列图书一样, 本书着重强调了 Visual Basic 程序设计中一步一步解决问题的方法。书中的每一节都遵循一致的风格, 引导你从具体的问题一步步走向解决问题的办法, 每一节都包含解决一个问题的具体步骤以及一个它们是怎样工作和为什么工作的讨论。在大多数情形下, 你只须把书中提供的代码和对象拷贝到你的应用程序中, 它们便能立即运行。本书中的所有代码在书后的 CD-ROM 上都能找到。

这本书中的概念和例子对各种层次的 Visual Basic 程序员都是非常有用的。每一节都按复杂程度分级, 同时带有大量的有关信息以满足读者的需要。

使用本书需要什么

你需要 Visual Basic 6.0 的专业版或企业版。这本书是根据企业版编写的, 但大部分内容也适用于专业版。有些地方还适用于 Visual Basic 5.0, 只是一些特定的菜单选项和窗口随版本的不同有些变化。因此要使这些例子在 Visual Basic 5.0 中工作, 需要适当地做一些修改。

本书的大多数章节都避免使用 Visual Basic 6.0 专业版或企业版中没有的控件或工具, 但尽管如此, 我们应该看到 Visual Basic 的一个非常重要的特点在于它使用第三方控件和工具的扩展性上。因此建议你积极探索第三方厂商提供的这些工具和控件, 它们常常能大大地缩短你的开发周期。

目 录

第 1 章 使用绑定控件访问数据库	(1)
1.1 使用绑定控件浏览记录集.....	(2)
1.2 输入合法的数据到绑定控件中.....	(5)
1.3 允许用户撤销绑定控件中的操作.....	(8)
1.4 使用绑定控件增加和删除记录.....	(12)
1.5 创建和使用绑定列表.....	(17)
1.6 显示一个主记录的许多详细记录.....	(24)
1.7 从代码中改变数据绑定网格单元中的数据.....	(29)
1.8 恰当地处理数据库错误.....	(32)
第 2 章 使用数据访问对象访问数据库	(39)
2.1 使用数据访问对象浏览和更新记录集.....	(39)
2.2 验证输入到数据访问对象的数据.....	(52)
2.3 允许用户撤销它们在数据访问对象里所做的改变.....	(56)
2.4 使用数据访问对象添加和删除记录.....	(59)
2.5 使用非绑定控件来更新数据访问对象的字段.....	(66)
2.6 使用数据访问对象的索引值查找记录.....	(71)
2.7 确定动态集类型或快照类型的记录集中有多少记录.....	(78)
2.8 处理数据访问对象中的错误.....	(81)
2.9 使用数据访问对象访问 Excel 工作表	(86)
第 3 章 使用 SQL 创建查询	(95)
3.1 从单个表选取记录来创建记录集.....	(96)
3.2 在 SQL 查询中选取唯一字段的值.....	(101)
3.3 在 SQL 查询中使用变量和 Visual Basic 函数	(106)
3.4 在 SQL 查询中使用通配符和值范围.....	(110)
3.5 定义和使用参数查询	(113)
3.6 利用多个表的字段创建新的记录集	(117)
3.7 在一个没有相关表对应条目的表中查找记录	(120)
3.8 检索计数、平均和求和等信息并通过把它绑定到数据控件中来显示它.....	(123)
3.9 创建一个可以包含重复记录的记录集	(127)
3.10 在 SQL 语句中使用 Visual Basic 函数	(132)
3.11 对数据库中的记录做大量的更新.....	(134)
3.12 创建和删除表.....	(138)
3.13 添加和删除记录.....	(149)

3.14	用已存在的表中的数据创建新表	(158)
3.15	修改表的结构	(162)
3.16	创建一个交叉表的查询	(170)
第4章	设计和实现数据库	(174)
4.1	创建新数据库	(174)
4.2	定义表和字段	(178)
4.3	定义主键和其他索引	(193)
4.4	定义表间的关系	(205)
4.5	使用 Jet 数据库引擎加强商业规则	(216)
第5章	Microsoft Access 数据库	(223)
5.1	确定表中记录的个数及创建和最后修改的日期	(223)
5.2	从其他数据库文件连接表	(229)
5.3	输入一个文本文件	(236)
5.4	在数据库文件中存储图形	(250)
5.5	保证多表操作的完整性	(255)
5.6	压缩及修复数据库	(263)
5.7	使用保存在 Microsoft Access 数据库中的参数查询	(266)
第6章	连接 ODBC 服务器	(274)
6.1	使用 ODBC 管理器维护数据源	(280)
6.2	通过 Visual Basic 数据控件使用 ODBC	(283)
6.3	使用 RegisterDatabase 创建可访问 ODBC 的数据源	(286)
6.4	打开 ODBC 数据库时不显示注册对话框	(293)
6.5	确定 ODBC 服务器提供何种服务	(301)
6.6	使用 ODBCDirect 连接数据库服务器	(317)
第7章	SQL Server 数据库和远程数据对象	(321)
7.1	用远程数据控件浏览 SQL Server 数据库	(322)
7.2	使用远程数据控件在 SQL Server 数据库中增加和删除记录	(325)
7.3	使用远程数据对象连接 SQL Server 数据库	(329)
7.4	使用远程数据对象浏览 SQL Server 数据库中的记录	(331)
7.5	使用远程数据对象在 SQL Server 数据库中增加、更新和删除记录	(337)
7.6	使用远程数据对象执行一个 SQL Server 存储过程	(347)
7.7	使用远程数据对象执行一个参数化的 SQL Server 存储过程	(349)
7.8	处理远程数据对象错误	(353)
第8章	ActiveX 数据对象	(358)
8.1	用 ADO 数据控件浏览 SQL Server 数据库	(359)
8.2	用 ADO 数据控件在 SQL Server 数据库中创建和删除记录	(361)
8.3	用 ActiveX 数据对象从 SQL Server 中检索数据	(364)

8.4	用 ActiveX 数据对象改变数据	(367)
8.5	用 ActiveX 数据对象进行事务处理	(376)
8.6	用 ActiveX 数据对象执行一个 SQL Server 存储过程	(380)
8.7	用 ActiveX 数据对象执行一个参数化的 SQL Server 存储过程	(384)
8.8	用 ActiveX 数据对象创建和修改 SQL Server 对象	(387)
8.9	用 ActiveX 数据对象执行成批的更新	(389)
8.10	用 ActiveX 数据对象进行远程数据更新	(394)
8.11	用 ActiveX 数据对象创建一个中间层商业对象	(396)
8.12	在 Microsoft 事务处理服务器中封装商业对象	(405)
8.13	处理 ActiveX 数据对象中的错误	(409)
第 9 章	Microsoft Data Report	(414)
9.1	用数据报表设计器创建报表	(418)
9.2	给数据报表加上计算字段	(422)
9.3	用 Visual Basic 代码来控制数据报表的显示、打印或输出	(425)
9.4	根据选择的准则创建数据报表	(431)
9.5	用 Crystal Reports 确定要打印的记录	(434)
9.6	用 Crystal Reports 创建小计和其他计算字段	(441)
9.7	用 Crystal Reports 控制记录的打印顺序	(452)
9.8	用 Crystal Reports 打印标签	(455)
9.9	用 Crystal Reports 生成和打印窗体信件	(458)
9.10	用 Crystal Reports 实现字段间没有多余空格	(464)
9.11	用 Crystal Reports 避免不含数据的字段打印时出现空行	(468)
9.12	用 Crystal Reports 创建交叉表格式报表	(473)
9.13	根据用户输入的变量生成报表	(477)
第 10 章	数据库安全和多用户访问	(484)
10.1	打开一个数据库当用户正在使用它时其他用户不能访问	(484)
10.2	打开一个表当用户正在使用时其他用户不能访问	(489)
10.3	使用锁定的记录	(496)
10.4	使用安全的 Microsoft Access 数据库文件	(504)
10.5	为数据库对象分配许可	(511)
10.6	改变数据库对象的所有关系	(522)
10.7	修改或删除数据库口令	(526)
10.8	所有用户用一个口令访问数据库	(531)
10.9	向系统数据库中加入新的用户	(536)
10.10	在系统数据库中定义一个新的用户组	(541)
10.11	在用户组中添加和删除用户	(545)
10.12	跟踪用户在数据库中的活动	(551)
10.13	创建和使用一个加密数据库	(556)

第 11 章 Windows 注册表和状态信息	(562)
11.1 从 Visual Basic 中输入和检索 Windows 注册表项	(562)
11.2 把与数据访问相关的信息放入注册表	(567)
11.3 确定随应用程序一同发布的相关的数据库和报表文件	(574)
11.4 通过注册表项调整 Jet 引擎	(580)
11.5 使用注册表项调整 ODBC 引擎	(583)
第 12 章 ActiveX 和 Automation	(587)
12.1 用 ActiveX Automation 编辑嵌入在应用程序中的数据库里 的图像、文档和电子表格	(589)
12.2 通过私有类模块访问数据库文件	(596)
12.3 发布 ActiveX 私有类以便其他人可以使用它	(610)
12.4 用 ActiveX EXE 服务器控制访问加密口令保护的数据库	(612)
12.5 把 Microsoft Access 作为一个 ActiveX 服务器打印报表	(617)
12.6 在 Visual Basic 中使用一个带有用户定义函数的 Access 查询	(619)
12.7 把数据库中的数据放到电子表格中进行统计分析	(627)
12.8 在一个 VB 应用程序中包含一个可编辑的报表书写器	(634)
12.9 ActiveX 组件与注册表	(643)
12.10 创建自己的定制数据控件	(645)
12.11 创建自己的数据绑定控件	(665)
12.12 使用带有定制数据绑定控件的 DataRepeater 控件	(672)
第 13 章 数据库高级技术	(676)
13.1 用 Soundex 算法查找数据库记录	(676)
13.2 备份选择的数据库对象	(679)
13.3 用 Jet 引擎复制一个数据库	(687)
13.4 复制过程中忽略指定的对象	(693)
13.5 创建一个复制数据库的非复制版本	(696)
附录 A SQL 语言参考	(699)
附录 B 数据访问对象(DAO)参考	(710)
附录 C 远程数据对象(RDO)参考	(725)
附录 D ActiveX 数据对象(ADO)参考	(735)
附录 E Visual Basic 对象简介	(742)

第1章 使用绑定控件访问数据库

- 1.1 使用绑定控件浏览记录集
- 1.2 输入合法的数据到绑定控件中
- 1.3 允许用户撤销绑定控件中的操作
- 1.4 使用绑定控件增加和删除记录
- 1.5 创建和使用绑定列表
- 1.6 显示一个主记录的许多详细记录
- 1.7 从代码中改变数据绑定网格单元中的数据
- 1.8 恰当地处理数据库错误

Visual Basic 提供的 Microsoft Jet 数据库引擎使你有能力访问多种类型的数据库——Microsoft Access 数据库;其他的基于 PC 机的数据库(如 dBASE, FoxPro, Paradox 和 Btrieve);以及任何支持开放性数据库互连(ODBC)标准的关系型数据库。Visual Basic 提供了两种使用 Jet 数据库引擎的基本方法:数据控件和数据访问对象(DAO)。数据控件需要较少的代码,但是数据访问对象要灵活得多。本章将告诉你如何使用数据控件去执行基本的数据库操作。第2章“使用数据访问对象访问数据库”将描述数据访问对象的使用方法。

VB 专有术语入门

如果你是数据库编程的新手,可能有许多不认识的 VB 术语。VB 对数据库的工作通过一个记录集来完成,该记录集包含一个表中的所有记录或满足特定结构化查询语言(SQL)的 SELECT 语句的所有记录。一个 SELECT 语句从一个或多个符合特定条件的数据库表中指定数据库字段。我们将在第3章“使用 SQL 创建查询”中讨论 SQL。

编程人员与用户的交互通过放在窗体上的可视化控件进行,包括数据输入、命令按钮、菜单、标签、列表框等。最普通的控件是用来输入数据的文本框、使程序有效工作的命令按钮、菜单以及用来描绘其他控件的标签。列表框和组合框为用户提供文本输入的多种选择。

大多数可视化控件,包括文本框、列表框和组合框可以绑定到数据源上自动显示数据,或者有特殊的数据绑定窗口。绑定是连接可视化控件与记录集中的字段的过程。最普通的绑定方法是数据控件。数据控件有一个可视化界面以支持数据在记录之间移动,还有一个记录集对象管理到数据库引擎的接口。数据控件组件也支持编程和设计时控件的方法和属性。组件只是用来创建一个 VB 应用程序的微小部分。使组件有效工作的方法等同于对组件的函数调用。属性是有助于控制组件行为的组件数据元素。例如,数据控件有一个 DatabaseName 属性,它指定数据库的位置,还有一个 Move 方法,可以在窗体上移动可视化控件。此外,数据控件显示它所包含的记录集对象的所有操作和属性。

本章所有的例子使用随 Visual Basic 一同发布的现有的 Microsoft Access 数据库文件(后面章节将阐述如何通过 VB 创建数据库)。但是,本章介绍的这些技术适用于 VB 通过 Jet 引擎访问的所有数据库。此外,企业版的远程数据控件(RDO)也使用类似的技术直接使用 ODBC 数据库。远程数据控件绕开 Jet 引擎,通常它比通过引擎访问更快。

查找例子

本书的所有内容均在随机附带的 CD-ROM 中。当你安装完源代码之后,会看到每一章都有单独的目录。从你的安装目录中将有机会预览每一章的步骤。如果你想单独运行一章,我们建议你在你的计算机上的不同工作区运行。

1.1 使用绑定控件浏览记录集

问题

我想查看一个数据库中的记录,但我不想编写很多代码,在 Visual Basic 中如何实现?

技术

Visual Basic 的数据控件对象,与数据绑定控件一起,可以使你不用编写任何代码,就可以浏览一个数据库中的记录。

要使用数据控件,首先要把它放在你的窗体上,并设置两个属性:第 1 个属性 DatabaseName,指明它所链接的数据库,第 2 个属性 RecordSource,设置数据库内的数据源。对每一个你想访问的 RecordSource 数据库字段,在窗体上添加一个文本框,并将它们与数据控件对象和 RecordSource 字段绑定在一起。

兼容的数据库

与 Visual Basic 数据控件以及在第 2 章讨论的 Visual Basic 数据访问对象兼容的数据库,包括 Microsoft Access, dBASE, FoxPro, Paradox, Btrieve 和其他支持 ODBC 标准的数据库产品。大部分桌面系统及多用户系统的关系数据库产品支持 ODBC。本书中的示例使用 Microsoft Access 数据库,但第 6 章“连接 ODBC 服务器”和第 7 章“SQL Server 数据库和远程数据对象”与其他数据库产品有关。除这两章外的所有例子几乎适用于其他所有数据库。

当你使用 Microsoft Access 数据库时,DatabaseName 就是 Microsoft Access 数据库文件的名称。当你使用其他数据库产品时,取决于数据库的类型(如 dBASE, FoxPro 和 Paradox 数据库),DatabaseName 就是数据文件所在的目录。RecordSource 可以是一个表名也可以是 SQL 的 SELECT 语句。Microsoft Access 也允许你将一个数据库内的查询名称指定为 RecordSource。

数据控件不仅在你的窗体和数据库之间提供链接,同时它还提供了在数据库中导航的工具。图 1.1 显示了一个数据控件,Next Record 和 Previous Record 按钮在数据库中一次移动一条记录,First Record 和 Last Record 按钮则可快速移到数据库的开始或结尾。

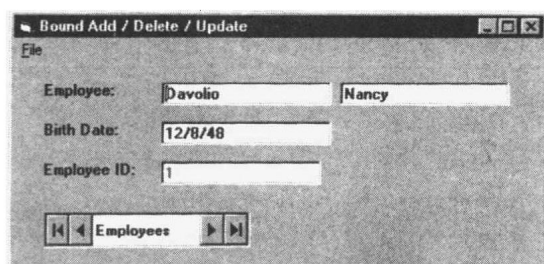


图 1.1 数据控件

步骤

要预览本节的内容,打开 CD-ROM 中 Chapter01\HowTo01 目录下的 BrowseBound.VBP 项目。将数据控件 datEmployees 的 DatabaseName 属性指向你系统中(与 VB6.EXE 在同一个目录下)安装的 NWind.MDB。然后运行项目就会出现如图 1.2 所示的窗体。使用数据控件上的按钮查看 NWind.MDB 的 Title 表中的记录。

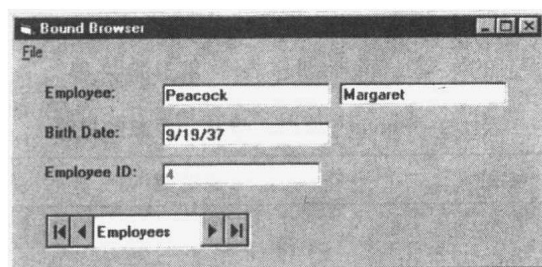


图 1.2 Bound Browser 窗体

1. 创建一个新项目 BrowseBound.VBP。按表 1.1 向 Form1 中添加对象并设置其属性,将窗体保存为 BrowseBound.FRM。修改 datEmployees 的 DatabaseName 属性使其路径指向 NWIND.MDB。

表 1.1 Bound Browser 窗体的对象及其属性

对象	属性	设置
Form	Name	Form1
	Caption	"Bound Browser"
Data	Name	datEmployees
	Caption	"Employees"
	DatabaseName	"D:\Program Files\Microsoft Visual Studio\VB6\NWIND.MDB"
	RecordSource	"Employees"
TextBox	Name	txtEmpLastName
	DataField	"LastName"
	DataSource	"datEmployees"

(续表)

对象	属性	设置
TextBox	Name	txtEmpFirstName
	DataField	"FirstName"
	DataSource	"datEmployees"
TextBox	Name	txtBirthDate
	DataField	"BirthDate"
	DataSource	"datEmployees"
TextBox	Name	txtEmployeeId
	DataField	"EmployeeID"
	DataSource	"datEmployees"
	Enabled	False
Label	Name	Label1
	Caption	"Employee:"
Label	Name	Label2
	Caption	"Birth Date:"
Label	Name	Label3
	Caption	"Employee ID:"

2. 利用 Visual Basic 中的菜单编辑器创建表 1.2 中的菜单。

表 1.2 Bound Browser 窗体上菜单的描述

标题	名称	快捷键
&.File	mnuFile	
---E&.xit	mnuFileExit	

3. 为 mnuExit 的 Click 事件添加如下代码：

```
Private Sub mnuFileExit_Click()
    Unload Me
End Sub
```

工作过程

当应用程序开始执行时,数据控件打开 NWind.MDB 数据库,在 Titles 表中生成一条记录,并显示窗体上绑定控件的记录集中的第一条记录。Visual Basic 中的记录集(recordset)对象是用来操作数据库的内容的。绑定控件是一些可视化的界面控件,人们可以在屏幕上看到它们,而且它们也与数据控件的记录集所管理的字段链接或绑定在一起。记录集提供记录之间的移动、添加、更新以及删除的方法。当用户单击数据控件的记录导航按钮时,数据控件记下选中记录的位置,并用新记录的值更新绑定的控件。

在内部,数据控件的工作过程很复杂。你将看到带有文本框的屏幕窗体。图 1.3 显示了绑定的文本框、数据控件和数据控件记录集之间的交互。当数据控件移动到一个不同的记录时,它就检查绑定文本框和记录集字段的数据变化。如果发现有改变,数据控件将数据移动到相应的字段并自动更新数据库。最后,数据控件检索到所需的记录并将各个字段拷贝到文

本控件来显示。本章余下的部分将探讨数据控件的事件和方法,用较少的工作来创建纯粹的应用程序。

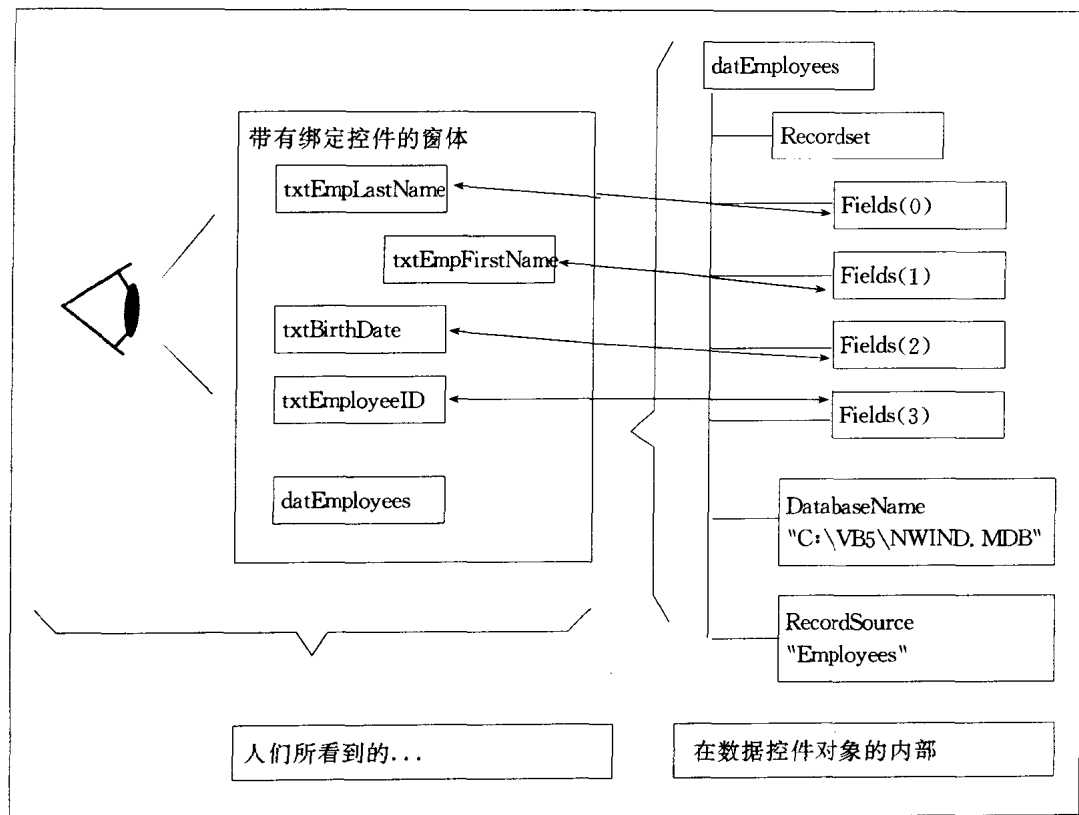


图 1.3 在数据控件对象的内部

注释

这是真正不用代码的开发。唯一一条可执行的代码行关闭应用程序。但这只是非常有限的应用程序,还没有对添加记录及删除记录做有效性的检查。要执行这些操作,必须要有一些代码——但并不是需要许多代码,而只是一些。以下各节将显示如何添加这些功能。

1.2 输入合法的数据到绑定控件中

问题

数据控件和绑定控件提供少量代码的数据库访问。但我需要在更新数据库之前,检验输入到窗体中的数据的有效性。使用绑定控件时,我如何检查输入数据的有效性呢?

技术

每次更改与一个数据控件相连的记录集中的当前记录时(通过移动记录、删除当前记录或关闭记录集),VB 都会触发数据控件的 Validate 事件。你可以编写一个事件子程序来检查

绑定控件中的数据改变。

Validate 事件子程序接受两个参数：

- **Action** 描述引发 Validate 事件的整数。
- **Save** 一个布尔量。如果绑定控件中的数据已发生变化则为 True；如果数据没有变化，则为 False。

在事件子例程中，可以检查 Save 的值。如果是 True，那就可以检查每一个值是否合法。如果有非法的值，可以将 Action 置为内置的常数 dbDataActionCancel，这将会取消引发 Validate 事件的事件。例如，当单击数据控件来移动到不同的记录而触发 Validate 事件时，将 Action 置为 dbDataActionCancel 取消 Move 事件并将数据控件定位在原来的记录上。Validate 事件子程序也应该显示问题的消息框以使输入被校正。

步骤

打开 Chapter01\HowTo02 目录下的 ValidateBound. VBP 项目，就可以预览本节的结果。将 datEmployees 数据控件的 DatabaseName 属性改为 NWind. MDB(与 VB6. EXE 在同一个目录下)，然后运行项目，就会显示出与图 1.2 类似的窗体。用数据控件上的按钮查看 NWind. MDB 中的 Employee 表的记录。选中 Employee 和 Birth Date 文本框中的所有文字，并删除它们，然后试着移动到下一条记录。你将看到如图 1.4 所示的错误消息，告诉你必须输入姓、名和生日。选择 File|Exit 菜单选项，关闭项目。

1. 在你的工作区中创建一个新的名为 ValidateBound. VBP 的项目。并用 Form1 生成原表 1.1 中的对象和属性，并将窗体保存为 ValidateBound. FRM(与 1.1 节中的窗体一样)。将 datEmployees 的 DatabaseName 属性变为 NWind. MDB。用 Visual Basic 的菜单编辑器生成表 1.2 所示的菜单。
2. 按下 Ctrl+D 键或从主菜单中选择 Project|Add File 选项，将 Chapter01\HowTo02 目录下的 clsUtility.cls 文件加入到你的项目中。利用 File 通用对话框选择文件。

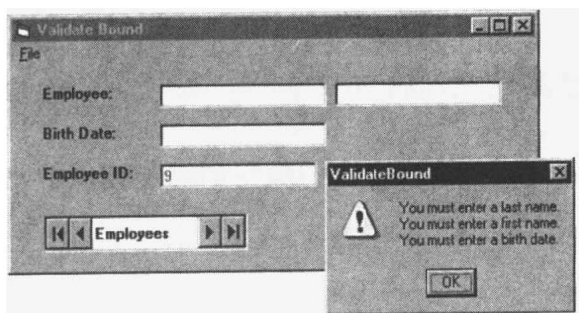


图 1.4 Validate Bound 的窗体

3. 将以下代码添加到 Form1 的声明部分。Utility 类用来连接 MsgBox 中的字符串。

```
Private Utility As New clsUtility  
Private mblnValidationFailed As Boolean
```

4. 在 Form1 中为 datEmployees 数据控件添加 Validate 事件处理代码。这段代码用来检

查所键入的数据的有效性。如果有非法数据,子程序显示错误消息,取消 Validate 事件,并将窗体级的变量 mblnValidationFailed 置为 True。

```
Private Sub datEmployees_Validate(Action As Integer, Save As Integer)
    Dim strMsg As String
    Dim enumMsgResult As VbMsgBoxResult

    If Save = True Or Action = vbDataActionUpdate _
        Or Action = vbDataActionUnload Then
        ' One or more bound controls has changed or a previous
        ' validation failed, so verify that all fields have legal
        ' entries. If a field has an incorrect value, append a
        ' string explaining the error to strMsg and set the focus
        ' to that field to facilitate correcting the error. We
        ' explain all errors encountered in a single message box.
        strMsg = " "
        If txtEmpLastName.Text = " " Then
            Utility.AddToMsg strMsg, "You must enter a last name."
            txtEmpLastName.SetFocus
        End If
        If txtEmpFirstName.Text = "" Then
            Utility.AddToMsg strMsg, "You must enter a first name."
            txtEmpFirstName.SetFocus
        End If
        If Not IsDate(txtBirthDate.Text) Then
            Utility.AddToMsg strMsg, "You must enter a birth date."
            txtBirthDate.SetFocus
        Else
            If CDate(txtBirthDate.Text) >= Date Then
                Utility.AddToMsg strMsg, _
                    "Birth date must be in the past."
                txtBirthDate.SetFocus
            End If
        End If
        If strMsg <> "" Then
            ' We have something in the variable strMsg, which
            ' means that an error has occurred. Display the
            ' message. The focus is in the last text box where
            ' an error was found.
            MsgBox strMsg, vbExclamation
            ' Cancel the Validate event
            Action = vbDataActionCancel
            ' Deny form Unload until fields are corrected
            mblnValidationFailed = True
        Else
            mblnValidationFailed = False
        End If
    End If
End Sub
```

5. 将以下代码添加到 Form1 中作为 Unload 事件。如果 Validate 事件设置了 UpdateCancelled 标志,这段程序取消 Unload 事件。

```
Private Sub Form_Unload(Cancel As Integer)
```

```
' Don't allow the unload until the data is validated.  
If mblnValidationFailed Then Cancel = True  
End Sub
```

6. 将以下代码添加到 mnuExit 的 Click 事件中。

```
Private Sub mnuExit_Click()  
    Unload Me  
End Sub
```

工作过程

每次 Validate 事件被调用时,每个控件中的内容就会被检查以确信它们是否包含合法数据。如果没有,Validate 事件就被取消。这会防止带无效数据的记录被保存。Validate 事件处理程序利用一个“helper”实用程序类来添加多条错误消息。一次显示多条错误对用户来说是一种很好的设计技术,因为这样可以减少用户的麻烦。

当窗体被卸出时,绑定控件的内容会自动地通过数据控件来保存,这就意味着 Validate 事件会被调用。如果控件中有非法数据,Validate 事件就被取消,但这不会导致取消 Form Unload 事件。因此,Validate 事件要设置一个窗体级的变量 mblnValidationFailed,这个变量在 Form Unload 过程中检查。如果 mblnValidationFailed 为 True,Form Unload 事件被取消,该应用程序也不会结束。

注释

有效性检查可以帮助控制数据输入错误,但它没有提供取消选项来恢复窗体的变化。当这个窗体的数据字段被改变时,用户必须输入有效的数据才能修改记录或退出应用程序。显然,应该有更好的解决办法,下面将接着介绍。

1.3 允许用户撤销绑定控件中的操作

问题

我希望窗体在记录被保存前可以恢复记录。如何用绑定控件实现这一目的?

技术

通过使用数据控件中的 UpdateControls 方法,你的窗体就会具有恢复能力。该方法使 Visual Basic 从数据库文件中重新读取当前记录,并将相应的数据控件中的数据刷新,只需执行这个方法,就可以用数据库中的初始数据重复绑定控件中的任何变化。

步骤

打开 UndoBound.VBP 项目。将 datEmployees 数据控件的 DatabaseName 属性改为 NWind.MDB(与 VB6.EXE 在同一个目录下)。然后运行项目,就会显示如图 1.5 所示的窗体。用数据控件上的按钮来查看 NWind.MDB 中的 Employees 表中的记录。改变一条记录,

在你移动到另一条记录前,选择 Edit|Undo。你将看到所做变化被取消。

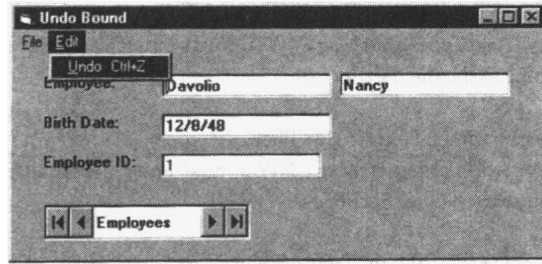


图 1.5 Undo Bound 窗体

1. 创建一个新的名为 UndoBound.VBP 的项目。并在 Form1 中创建表 1.1 列出的对象和属性,将窗体保存为 UndoBound.FRM(这与 1.1 节和 1.2 节中的窗体一样)。将 datEmployees 的 DatabaseName 属性变为 NWind.MDB。用 Visual Basic 的菜单编辑器创建表 1.3 所示的菜单。

表 1.3 UndoBound.FRM 菜单的描述

标题	名称	快捷键
&File	mnuFile	
----E&xit	mnuFileExit	
&Edit	mnuEdit	
----&Undo	mnuEditUndo	Ctrl+Z

2. 按下 Ctrl+D 键或从主菜单中选择 Project|Add File,将 Chapter01\HowTo03 目录下的 clsUtility.cls 文件加入到你的项目中。利用 File 通用对话框选择文件。
3. 将以下代码添加到 Form1 的声明部分。

```
Private Utility As New clsUtility
Private mblnValidationFailed As Boolean
```

4. 在 Form1 中为 datEmployees 数据控件添加 Validate 事件处理代码。(其代码与 1.2 节不同之处用黑体字标出)这段代码用来检查所键入的数据的有效性。如果有非法数据,子程序显示错误消息,并询问确定(OK)还是取消(Cancel)。一个 OK 响应取消 Validate 事件,并将窗体级的变量 mblnValidationFailed 置为 True;一个 Cancel 响应从数据库中重新检索数据,并将绑定窗体控件的值恢复。

```
Private Sub datEmployees_Validate(Action As Integer, Save As Integer)
    Dim strMsg As String
    Dim enumMsgResult As VbMsgBoxResult

    If Save = True Or Action = vbDataActionUpdate -
        Or Action = vbDataActionUnload Then
        ' One or more bound controls has changed or a previous
        ' validation failed, so verify that all fields have legal
        ' entries. If a field has an incorrect value, append a
        ' string explaining the error to strMsg and set the focus
```