Visual Basic 的数据库功能及应用

吴淑菊 侯晓兵 林碧琴 (北方交通大学通信与控制工程系 100044)

摘要:本文介绍了如何利用 Visual Basic 开发数据库软件,详细讨论了 Data 控件的各种操作以及利用 SQL 实现更加复杂的数据库操作。

关键词:Data 控件 属性 SQL

一、概述

Visual Basic 一个突出的新特性是增加了一个 Data 控件,它采用事件驱动编程机制,可以实现可视的数据存取,不编写代码就可以创建数据浏览应用程序,或者用程序进行控制,实现更多的功能。VB 的另一个数据控件为 Grid 控件,它是一个显示框的二维数组或单元,适合于显示数据库或电子表格信息。本文介绍如何使用 Data 控件以及如何利用它实现对数据库进行各种操作。

值得注意的是:要想在 Visual Basic 中使用数据,必须在 Windows 中实现文件共享,否则需要退出 Visual Basic 和 Windows,把 Share 加到 AUTOEXEC.BAT 文件中。

二、Data 控件及与数据显示控件的联系

1. Data 控件

Data 控件可以实现对数据库特定数据的存取。简单地说,就是通过设置数据控件的属性,把数据控件挂在一个数据库或数据库的某一个表上,通过联系显示控件与数据控件,在显示控件中显示响应的数据资料。一个应用程序中可以有一个或多个 Data 控件,每个控件控制一个数据集。利用 Data 控件可以访问如下的数据:

Microsoft Access

Microsoft FoxPro

Borland dBase

Borland Paradox

对 Data 控件的操作,最简单的是使用它的属性窗口,属性窗口控制了 Data 控件的外观和行为,在 Windows 菜单中选择 Properties 命令,按 F4 或选择工具条上的 Properties Window 按钮,都可激活属性窗口,在属性窗口中,可以定义如下一些与数据库操作有关的属性,其他属性可以由用户自行选择定义:

属性:	设置:
Connect:	这一属性的设置告诉 Visual Basic 将使用的数据库的种类,如果把属性设置成空字符串,表明数据库为 Microsoft Access 格式。其他种类的数据库的设置值为 "FoxPro2.5", "Paradox", "dBase Ⅳ",用户可以根据需要进行填写。
DatabaseName:	这一属性的设置告诉 Visual Basic 将要访问的数据库文件。用户应指明文件的路径。用户可以单击这一属性中的省略号,将显示一个普通对话框,用户可以选择所需的数据库文件。
RecordSource:	通常一个数据库中可以包含多个不同数据内容的表,这一属性的设置指明 Data 控件从哪一个表中获取数据。

2. 建立数据显示控件与 Data 控件的联系

利用 Data 控件,应用程序可以从数据库中取出足够的信息,再利用 Visual Basic 中的一些数据显示控件,如 Label, Text Box, Check Box, Image, Picture Box 等就能实现数据的显示与修改。其中 Label 和 Text Box 控件用来显示数字和字符信息,Check Box 显示一个 Boolean 值。Image 与 Picture Box 用于数据库存储画面的显示。这些控件通过下面的属性与 Data 控件取得联系。

属性:	设置:
DataSource:	这一属性告诉 VB 显示控件的数据来源于哪一个 Data 控件,因为一个应用程序中可以包含多个 Data 控件。在本属性中填的是 Data 控件的属性窗口中属性 Name 的设置值。
DataField:	这一属性表示本显示控件中显示的是记录的哪一个字段,这样通过 Data 控

件与显示控件，我们就可以在应用程序中显示数据库的内容，甚至可以通过应用程序对数据进行编辑，将结果送还原数据库，这就实现了对数据库的存取。

建立数据显示控件与 Data 控件的联系后，通过单击 Data 控件的箭头，可以把记录移到表的起始端、表的上一条记录、表的下一条记录、表的末尾上去。用户每按一次这些箭头键，相关的显示控件都会自动地更换成新的记录数据，因此用户不需要编写任何代码。

三、程序化控制、SQL 及实例

1. 数据的程序化控制及实例

虽然用户不需要编写任何代码，即可通过 Data 控件自身所具备的功能完成对记录项的移动，然而这种功能毕竟是有限的，在很多情况下，简单的记录项移动不能满足用户的需要，因此，我们可以通过 VB 对 Data 控件的程序扩展编写代码，在程序中来完成更加强大的功能。

程序控制下的 Data 控件除了可以通过代码改变在 Properties 窗口中设置的各种属性以外，在运行期间，代码还可以实现超出设计控件本身的功能：

Data1.Recordset.MoveFirst：

将当前位置记录移到表的起始端。

Data1.Recordset.MovePrevious：

将当前位置移到表的上一条记录。

Data1.Recordset.MoveNext：

将当前位置移到表的下一条记录。

Data1.Recordset.MoveLast：

将当前位置移到表的末尾上去。

以上四种方法相当于分别单击 Data 控件的四个箭头。

Data1.Refresh：

打开一个数据库，如果用户在程序中改变了 DataBaseName、Readonly、Exclusive 和 Connect 属性中的一个，则必须使用这一方法重新打开数据库。Refresh 方法将把数据库中第一条记录设置为当前记录项。

Data1.Recordset.AddNew：

把当前记录指针移到表的末尾并为用户清除一个拷贝缓冲区。用户可以借此向数据库中的表发送一个新的信息，不过 AddNew 方法并非真正在数据库中添加了一条信息，要想实现此功能，还要使用下面的 Updata 方法。

Data1.Recordset.Update：

在使用了 AddNew 方法后使用本方法，真正实现把拷贝缓冲区中的内容存入表中。

Data1.Recordset.Delete：

删除记录集中的当前记录，这种方法一次删除一条记录，执行完此命令后，用户必须使用一个前面提到的 Move 方法把记录指针从被删除的记录上移走。

Data1.Recordset.FindFirst：

查找满足某一条件的第一条记录，类似的 Find 方法还有 FindNext、FindPrevious、FindLast 等。

除了以上方法之外，VB 还提供了其他方法，这些方法基本包括了数据库编程中的大多数操作。下面这段程序是我们开发的通信设备故障自动诊断系统中软件的一小部分，并作了简化处理。首先创建一个窗体 Form 和几个按钮 Command3D 以及若干数据显示控件，如文本框 Text 等，当然 Data 控件是必不可少的。在窗体装入事件 Form-Load()中，首先对 Data 控件初始化，设置相关属性，然后设置相应数据显示控件的有关属性并打开数据库。BOF 和 EOF 是两个只读属性，分别用来判断当前记录是否在第一条记录之前和是否在最后一条记录之后，例中用来判断数据库是否为空。最后，使用 AddNew 方法使程序处于新添记录项状态。几个 Command3D 控件分别用来完成向数据库存储当前记录项、创建新的记录项、删除当前记录项、根据输入的测试序号检索记录项等功能。程序如下：

```
Sub Form-Load ()
    ' 初始化 Data1 控件及数据显示控件(例中只选择其中一项),EXIST 为一全程变量
    data1.DatabaseName = "DATA.MDB"
    data1.RecordSource = "sys"
    text1.DataField = "测试序号"
    data1.Refresh
    If data1.Recordset.BOF = True And data1.Recordset.EOF = True Then EXIST = True
    If EXIST = False Then data1.Recordset.MoveLast
    data1.Recordset.AddNew
End Sub

Sub Command3D1-Click ()
    ' 输入完一条记录后,将此记录存盘并开辟新的缓冲区
    data1.Recordset.Update
    data1.Recordset.AddNew
```

```
End Sub

    Sub Command3D2-Click ()
    '修改完一条记录后,将当前记录存盘并移至下一条记录
    data1.Recordset.Update
    data1.Recordset.MoveNext
    If data1.Recordset.EOF = True Then
      data1.Refresh
      data1.Recordset.MoveLast
      x = MsgBox("已经到最后一个数据项!", 16)
    End If
End Sub

Sub Command3D3-Click ()
    '根据输入的测试序号检索相应的记录项,其中"测试序号"为库中一个字段名
    Dim jians $
    jians = InputBox("请输入记录的测试序号:", "序号检索")
    If jians = "" Then
      Exit Sub
    End If
    aa = "测试名称 = '" + jians + "'"
    data1.Recordset.FindFirst aa
    If data1.Recordset.NoMatch = True Then
      x = MsgBox("没找到要检索的记录,请检查测试序号是否正确", 16)
    End If
End Sub

Sub Command3D5-Click ()
    '删除当前记录项
    x = MsgBox("是否删除当前记录项?", 36)
    If x <> 7 Then
      data1.Recordset.Delete
      data1.Recordset.MoveNext
      If data1.Recordset.RecordCount = 0 Then
        command3d5.Enabled = False
        data1.Enabled = False
        Exit Sub
      End If
      If data1.Recordset.EOF = True Then
        data1.Recordset.MovePrevious
      End If
    End If
End Sub
```

2. 利用 SQL 实现复杂的数据访问

除以上所介绍的内容之外,Visual Basic 还支持结构化查询语言(SQL),它用来从表中按照用户提供的规则来选择出记录。SQL 可以在运行期间使用,来设置数据控件的 RecordSource 属性,这就允许用户通过程序来创建与数据控件相关的动态组。有关 SQL 的资料很多,这里举一个简单的例子。假如数据库中有一个名为 sys 的表,且表中的一个字段名为 name。用户在文本框 Text2 中输入一个或多个文字或字符,欲在 sys 表中查找出所有的在 name 字段中包括了文本框 Text2 内容的记录项,如果用户拥有 TrueGrid 控件并与 Data 控件相连,则所有满足条件的记录项将自动显示在 TrueGrid 上。

```
Sub Text2-KeyPress (keyascii As Integer)
    If keyascii = 13 Then
    '输入字符后,如果按下了回车键
      If Text2.Text = "" Then
    '如果 Text2 为空,则列出表中所有的记录项
        data1.RecordSource = "select * from sys"
        data1.Refresh
      Else
    '列出所有在 name 字段中包含 Text2 内容的记录项
        data1.RecordSource = "select * from sys where name like '*" + Text2.Text + "*'"
        data1.Refresh
      End If
    End If
End Sub
```

以上程序均已在 VB3.0 状态下调试通过。

参考资料

[1] 《The Visual Basic 3 for Windows Handbook》Gary Cornell 著,学苑出版社,1994.10

[2] 《运行 Visual Basic 3.0 for Windows》Ross Nelson 著,学苑出版社,1994.5

(来稿时间:1996 年 10 月)