

第一篇 数据库基本操作

Visual Basic 6.0（后面若无特殊说明，都简称为 VB）数据库技术的优势，突出地表现其对数据库的基本操作上。使用 Visual Basic 6.0 数据库技术，可以方便地与 Access、Paradox、Oracle、SQL Server 等所有主流数据库建立连接，进行数据显示、存取、动态编辑、分类查询、报表生成等操作，相关操作技术简洁明了，代码开发效率高，周期短，易于维护与升级。

本篇将重点介绍 Visual Basic 6.0 的数据库基本操作技术，涉及如何利用 ADO 对象，用 ListView 控件显示数据库里的字段，使用 MSFlexGrid 控件显示数据库信息，使用 CrystalReport 设计报表打印等技术及相关控件的使用，同时还为读者提供了 Visual Basic 6.0 数据库基本操作的一些常用技巧。

本部分主要有以下精彩内容：

- 用 ListView 控件显示数据库里的字段；
- 用图形表示数据库里的数据；
- CrystalReport 报表打印；
- 用 TreeView 控件显示来自 ADO 的 Grouping 数据。

实例 1 显示数据表的内容

实例效果

查看数据表内容的方法有很多种。本实例演示了用 Data 控件绑定数据库，显示一个数据表内容的方法。当程序启动时，数据库的所有表都显示在 ListBox 框里。单击框里的表，这时就会把该表的所有字段显示在 MSFlexGrid 控件中，如图 1-1 所示。

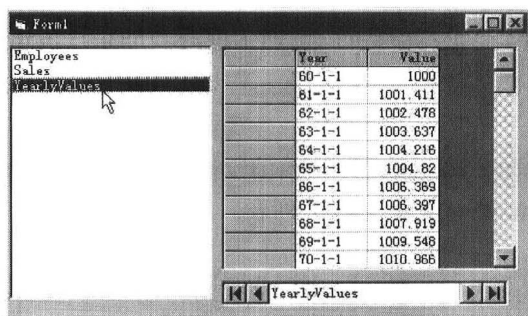


图 1-1 运行界面图

程序说明

本实例利用 SQL 查询语句获得表的信息。当选中某个表的时候，就从该表查询所有字段。实现代码如下：

```
table_name = List1.List(List1.ListIndex)
sql = "Select * From " & table_name
```

table_name 就是 ListBox 框里被选中的表，它是一个变量。

SQL 的基本操作如下。

1. 用 Select 子句检索记录

Select 子句是每一个检索数据的查询核心。它告诉数据库引擎返回什么字段。

Select 子句的常见形式是：

```
Select *
```

该子句的意思是“返回在所指定的记录源中能找到的所有字段”。这种命令形式很方便，因为你无需知道从表中检索的字段名称。检索表中的所有列是低效的，但如果只检索需要的字段，就可以大大地提高查询的效率。

2. 使用 From 子句指定记录源

From 子句说明的是查询检索记录的记录源；该记录源可以是一个表或另一个存储查询。你还能从多个表中检索记录，这在后面的章节中将介绍。

例如：

```
Select * From students
```

```
//检索 students 表中的所有记录
```

3. 用 Where 子句说明条件

Where 子句告诉数据库引擎根据所提供的一个或多个条件限定其检索的记录。条件是一个表达式，可具有真假两种判断。

例如：

```
Select * From students Where name="影子"
```

返回 students 中 name 字段为影子的列表，这次所返回的结果没有特定顺序，除非你使用了 Order By 子句。该子句将在后面的章节介绍。

注意：Where 子句中的文本字符串界限符是双引号，在 VB 中改为单引号，因为在 VB 中字符串的界定符是双引号。

此外，使用 And 和 Or 逻辑运算符可以将两个或更多的条件链接到一起，以创建更高级的 Where 子句。

例如：

```
Select * From students Where name="影子" And number>100
```

返回 name 为影子且 number 大于 100 的列表。

例如：

```
Select * From students Where name="影子" And (number>100 Or number<10)
```

返回 name 为影子、number 大于 100 或小于 10 的列表。

制作步骤

一、窗体界面的建立

1. 启动 Visual Basic 6.0，新建一个标准工程。
2. 向窗体中添加一个 MSFlexGrid 控件、一个 Data 控件和一个 ListBox 控件，详细设置请参阅本书光盘源代码。完成界面如图 1-2 所示。

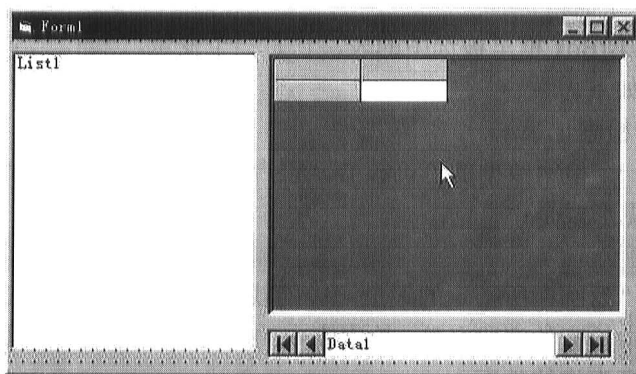


图 1-2 界面图

二、源代码与注释

```
Option Explicit
Private Sub Form_Load()
Dim dbname As String
Dim db As Database
```

```
Dim qdef As QueryDef
Dim td As TableDef

' 打开数据库
dbname = App.Path
If Right$(dbname, 1) <> "\" Then dbname = dbname & "\"
dbname = dbname & "data.mdb"
Set db = OpenDatabase(dbname)
' 在 ListBOX 框里显示表名
For Each td In db.TableDefs
    ' 不允许使用系统表
    If Left$(td.Name, 4) <> "MSys" Then _
        List1.AddItem td.Name
Next td
db.Close
' 连接数据库
Data1.DatabaseName = dbname
End Sub

' 选择表
Private Sub List1_Click()
Dim table_name As String
Dim sql As String
    table_name = List1.List(List1.ListIndex)
    sql = "SELECT * FROM " & table_name
    Data1.Caption = table_name
    Data1.RecordSource = sql
    Data1.Visible = True
' MSFlexGrid1.DataSource = Data1
' MSFlexGrid1.DataField = List1.List(List1.ListIndex)
    Data1.Refresh
    ' 使得 Data 、 DBGrid 控件可见
    MSFlexGrid1.Visible = True
End Sub
```

实例 2 利用 ADO 对象编程

实例效果

本例通过单击按钮，打开一个数据库，用 TreeView 控件显示该数据库的所有表，如图 2-1 所示。当选中某个表时，它就在 ListView1 和 ListView2 里显示所有的字段名和字段，如图 2-2、图 2-3 所示。

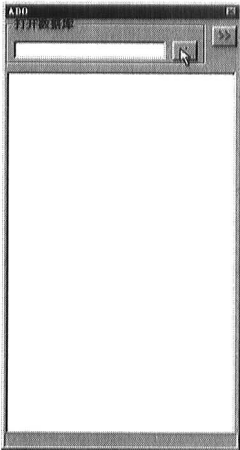


图 2-1 运行界面图

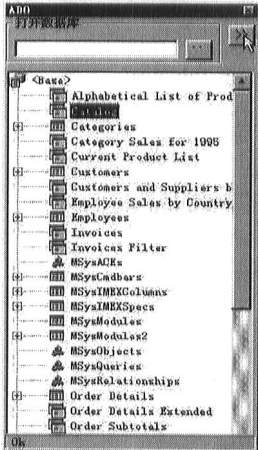


图 2-2 显示字段名

ProductID	Product Name	Quantity	Unit Price	Discount	Total Price
1	Chai	10	10.00	0.00	100.00
2	Chang	24	12.00	0.00	288.00
3	Aniseed Syrup	12	550.00	0.00	6600.00
4	Chief Cheese	40	9.00	0.00	360.00
5	Coriander	12	9.00	0.00	108.00
6	Coriander	12	1.10	0.00	13.20
7	Coriander	12	1.10	0.00	13.20
8	Coriander	12	1.10	0.00	13.20
9	Coriander	12	1.10	0.00	13.20
10	Coriander	12	1.10	0.00	13.20
11	Coriander	12	1.10	0.00	13.20
12	Coriander	12	1.10	0.00	13.20
13	Coriander	12	1.10	0.00	13.20
14	Coriander	12	1.10	0.00	13.20
15	Coriander	12	1.10	0.00	13.20
16	Coriander	12	1.10	0.00	13.20
17	Coriander	12	1.10	0.00	13.20
18	Coriander	12	1.10	0.00	13.20
19	Coriander	12	1.10	0.00	13.20
20	Coriander	12	1.10	0.00	13.20
21	Coriander	12	1.10	0.00	13.20
22	Coriander	12	1.10	0.00	13.20
23	Coriander	12	1.10	0.00	13.20
24	Coriander	12	1.10	0.00	13.20
25	Coriander	12	1.10	0.00	13.20
26	Coriander	12	1.10	0.00	13.20
27	Coriander	12	1.10	0.00	13.20
28	Coriander	12	1.10	0.00	13.20
29	Coriander	12	1.10	0.00	13.20
30	Coriander	12	1.10	0.00	13.20

图 2-3 显示字段

程序说明

本实例利用了 TreeView 控件的特性，把 TreeView 控件做成了动态添加数据的选单，使用起来非常方便、直观。增加了程序的趣味性。也让读者轻松掌握 TreeView 控件的使用，特别是树的动态使用。

本例充分利用了 ADO 对象。ActiveX Data Objects (ADO) 是微软公司最新的数据访问技术。它被设计用来同新的数据访问层 OLE DB Provider 一起协同工作，以提供通用数据访问 (Universal Data Access)。OLE DB 是一个低层的数据访问接口，用它可以访问各种数据源，包括传统的关系型数据库，以及电子邮件系统和自定义的商业对象。

ADO 向我们提供了一个熟悉的、高层的对 OLE DB 的 Automation 封装接口。对那些熟悉 RDO 的程序员来说,你可以把 OLE DB 比作是 ODBC 驱动程序。如同 RDO 对象是 ODBC 驱动程序接口一样,ADO 对象是 OLE DB 的接口;如同不同的数据库系统需要它们自己的 ODBC 驱动程序一样,不同的数据源要求使用它们自己的 OLE DB 提供者 (OLE DB provider)。目前,虽然 OLE DB 提供者比较少,但微软公司正积极推广该技术,并打算用 OLE DB 取代 ODBC。

ADO 具有很多优点,包括易于使用,熟悉的界面,高速度以及较低的内存占用(已实现 ADO 2.0 的 Msado15.dll 需要占用 342kB 内存,比 RDO 的 Msrdo20.dll 的 368kB 略小,大约是 DAO 3.5 的 Dao350.dll 所占内存的 60%)。同传统的数据对象层次 (DAO 和 RDO) 不同,ADO 可以独立创建。因此你可以只创建一个“Connection”对象,但是可以有多个独立的“Recordset”对象来使用它。ADO 针对客户/服务器以及 Web 应用程序作了优化。

制作步骤

一、窗体界面的建立

1. 启动 Visual Basic 6.0, 新建一个标准工程。
2. 向窗体中添加一个 Frame 控件, 在它上添加一个 TextBox 文本框和一个按钮; 向窗体中添加一个 CommandButton, 其属性中的 caption 设置为 “》”, 向窗体中添加一个 TreeView 和一个 ListView, 详细设置请参阅本书光盘源代码。界面如图 2-4 所示。

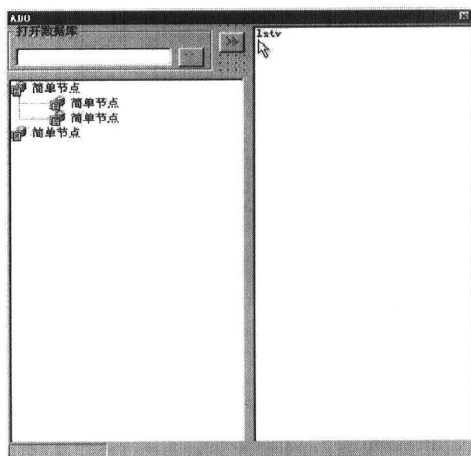


图 2-4 界面图

二、源代码与注释

```
Option Explicit
Dim Gconn As New ADODB.Connection '定义个新的 ADO 对象连接
Dim Rs As ADODB.Recordset '定义一个 ADO 对象数据集
Dim RsField As ADODB.Recordset
Dim MaTable As String
Dim Num As Long
Dim BaseConn As String
Dim ChaineConn As String
Private Sub Command1_Click()
```

```

Dim Node As Node
Dim etape1 As Node
Dim etape2 As Node
cd.Filter = "Base Access!*.mdb"
cd.ShowOpen
ChaineConn = BaseConn & cd.FileName
If cd.FileName = "" Then Exit Sub
trv.Nodes.Clear
Gconn.Open ChaineConn
Set Rs = Gconn.OpenSchema(adSchemaTables) '打开数据
Set Node = trv.Nodes.Add(, , "<Base>", 1, 1)
Node.Expanded = True
Do While Not Rs.EOF
    Select Case LCase(Rs.Fields(3))
        Case "table"
            Num = 2
        Case Is = "view"
            Num = 5
        Case "system table"
            Num = 6
        Case Else
            Num = 4
    End Select
    '增加在 TreeView 的子节点上
    Set etape1 = trv.Nodes.Add(Node, tvwChild, , Rs.Fields(2).Value, Num, Num)
    Set RsField = Gconn.OpenSchema(adSchemaColumns)
    Do While Not RsField.EOF
    If RsField.Fields(2) = Rs.Fields(2) Then
        '增加表格
        Set etape2 = trv.Nodes.Add(etape1, tvwChild, , RsField.Fields(3), 3, 3)
        End If
        RsField.MoveNext
    Loop
    Rs.MoveNext
Loop
Gconn.Close
End Sub
Private Sub Command2_Click()
Static Flag As Boolean

```

```

Flag = Not Flag
If Flag Then
    Me.Width = 12000
    Command2.Caption = "<<"
Else
    Me.Width = 3615
    Command2.Caption = ">>"
End If
End Sub

Private Sub Form_Load()
Me.Width = 3615
sb.Panels(1).Width = 5000
'设置数据库连接
BaseConn = "Provider=Microsoft.Jet.OLEDB.3.51;Persist Security Info=False;Data Source="
End Sub

Private Sub trv_NodeClick(ByVal Node As MSComctlLib.Node)
Dim rep As Long
Dim RsPrivate As ADODB.Recordset
Dim item As ListItem
Dim Requete As String
Dim H, G As Integer
Gconn.Open ChaineConn
MaTable = Replace(Node.FullPath, "<Base>\", "")
rep = InStr(MaTable, "\")
If rep > 0 Then
    MaTable = Left$(MaTable, rep - 1)
End If
Requete = "select  * from " & MaTable
On Error GoTo hdl
Set RsPrivate = Gconn.Execute(Requete)
lstv.ListItems.Clear
lstv.ColumnHeaders.Clear
For H = 0 To RsPrivate.Fields.Count - 1
    lstv.ColumnHeaders.Add , , RsPrivate.Fields(H).Name
Next
lstv.View = lvwReport
Do While Not RsPrivate.EOF
Set item = lstv.ListItems.Add( , , RsPrivate.Fields(0))
For G = 1 To RsPrivate.Fields.Count - 1

```

```

item.SubItems(G) = RsPrivate.Fields(G)
Next
RsPrivate.MoveNext
Loop
sb.Panels(1).Text = "Ok"
Gconn.Close
Exit Sub
hdl:
sb.Panels(1).Text = Err.Description    '在 Pannels 里显示错误
On Error Resume Next
Gconn.Close
End Sub

```

实例 3 一个基本的数据库程序

实例效果

本实例是一个基本的数据库程序，包括添加、搜索、删除和查找数据等功能，是一个入门级的程序。添加和搜索数据是使用的 InputBox 对话框来完成的。界面如图 3-1 所示。

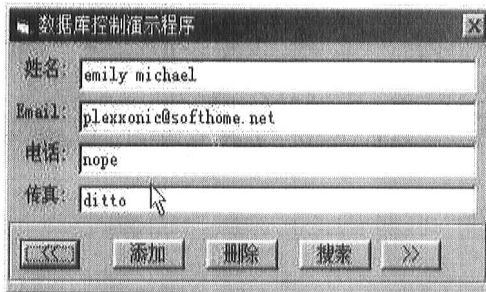


图 3-1 运行界面图

程序说明

本实例利用 DAO 对象。首先，定义了数据库和记录集，即 Dim db As Database、Dim rs As Recordset。再通过 OpenDataBase 打开数据库，通过 OpenRecordset 打开表。最后用 SQL 语句进行搜索。该程序没有用 Data、Adodc 等绑定数据库的控件。

当使用 DAO 访问 Access 2000 时可能会出现以下的错误信息：

"Run-time error 3343 Unrecognized Database Format XXX"

这个错误信息有两个解决的办法。

1. 如果使用 Data Control 连接 Access 2000 的数据库时，必须在 Data Control 的 Source 前先加上一行：

```
Set Data1.Recordset = rsDAO36 'rsDAO36 is a DAO 3.6
```

2. 如果是引用“Microsoft DAO 3.51 Object Library”来访问 Access 2000 的数据库时：在菜单的“工程/部件”下面加入“Microsoft DAO 3.6 Object Library”。

注意，以往在 VB 6 中一般是利用 ADO OLEDB Provider 3.51 访问 Access 97 程序，如果将 Access 97 升级到 Access 2000 之后，就无法顺利地使用 ADO OLEDB Provider 3.51 访问 Access 2000 数据库程序了。这种事情在 Microsoft 已经是司空见惯的事情了！任何软件只要有了新的版本，就会有和旧版本不相容的问题产生，因为 Access 2000 已经使用 Jet 4.0 Engine。解决的方法如下：必须将 ADO 的 Connect String 中的 Provider 修改为“PROVIDER=Microsoft,Jet.OLEDB.4.0”。

制作步骤

一、窗体界面的建立

1. 启动 Visual Basic 6.0，新建一个标准工程。

2. 向窗体中添加 4 个 TextBox 控件，其属性中的 Text 设置为空；向窗体中添加一个 Frame 控件，在它上面添加 5 个 CommandButton 控件，其属性中的 Caption 分别设置为“<<”、“添加”、“删除”、“搜索”、“>>”；向窗体中添加 4 个 Label 控件，其属性中的 caption 分别设置为“姓名”、“Email”、“电话”、“传真”，详细设置请参阅本书光盘源代码。界面如图 3-2 所示。

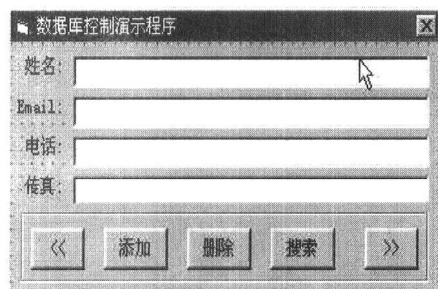


图 3-2 界面图

二、源代码与注释

```
Dim db As Database
Dim rs As Recordset
Private Sub cmdAdd_Click()
    newName = InputBox("Please Enter The Name To Add", "AddressBook")
    newEmail = InputBox("Please Enter The Email To Add", "AddressBook")
    newPhone = InputBox("Please Enter The Phone Number To Add", "AddressBook")
    newFax = InputBox("Please Enter The Fax Number To Add", "AddressBook")
    With rs
        .AddNew
        !Name = LCase(newName)
        !Email = LCase(newEmail)
        !Phone = LCase(newPhone)
        !Fax = LCase(newFax)
```

```
.Update
End With
MsgBox (newName & " Was Added To The Address Book")
End Sub

Private Sub cmdBack_Click()
On Error Resume Next
rs.MovePrevious
txtName = rs.Fields("Name")
txtEmail = rs.Fields("Email")
txtPhone = rs.Fields("Phone")
txtFax = rs.Fields("Fax")
End Sub

Private Sub cmdDelete_Click()
Set rs = db.OpenRecordset("SELECT ppl.name, ppl.email, ppl.phone, ppl.fax From ppl WHERE ppl.name = "
+ Chr$(34) + txtName.Text + Chr$(34) + ";") '设置查询
rs.MoveFirst '移动到第一个
Do Until rs.EOF
rs.Delete '删除
rs.MoveNext '移到下一个
Loop
db.Close
End Sub

Private Sub cmdForward_Click()
On Error Resume Next
rs.MoveNext
'赋值
txtName = rs.Fields("Name")
txtEmail = rs.Fields("Email")
txtPhone = rs.Fields("Phone")
txtFax = rs.Fields("Fax")
End Sub

Private Sub cmdSearch_Click()
Set rs = db.OpenRecordset("SELECT * FROM ppl")
NameQuery = InputBox("Enter A Name To Search For", "Name Query") 'INPUTBOX 对话框。
rs.MoveFirst
Do Until rs.EOF
```

```

If rs.Fields("name") Like "*" & LCase(NameQuery) & "*" Then
txtName = rs.Fields("Name")
txtEmail = rs.Fields("Email")
txtPhone = rs.Fields("Phone")
txtFax = rs.Fields("Fax")
Exit Sub
Else
rs.MoveNext
End If
Loop
End Sub

Private Sub Form_Load()
Set db = OpenDatabase(App.Path + "/ppl.mdb") '打开数据库
Set rs = db.OpenRecordset("ppl") '打开表
If rs.EOF Then '判断是否打开表
MsgBox "Please Add Someone To The Database"
Else
rs.MoveFirst
txtName = rs.Fields("Name")
txtEmail = rs.Fields("Email")
txtPhone = rs.Fields("Phone")
txtFax = rs.Fields("Fax")
End If
End Sub

Private Sub Label1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Label1.ToolTipText = txtName.Text
End Sub

Private Sub Label2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Label2.ToolTipText = txtEmail.Text
End Sub

```

实例 4 在 ListBox 中显示数据库内容

实例效果

当窗体启动时，本实例将自动连接到数据库，并把数据库中的内容显示在 ListBox 框里。单击 ListBox 框里每一条记录，这时将在另一窗体上显示它的字段，如有很多字段，还可单击

上一条、下一条、第一条和最后一条进行查看，如图 4-1 所示。

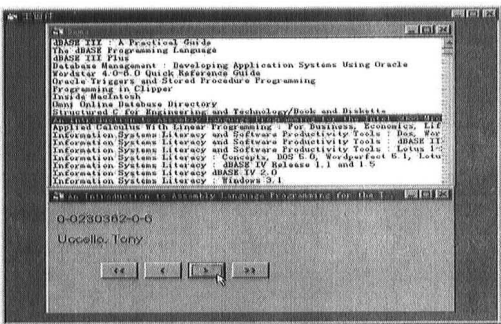


图 4-1 运行界面图

程序说明

本实例主要利用 SQL 查询语句，同时还用到主窗体和子窗体。程序启动时，把数据库的内容显示在 ListBox 框里。当选中一条记录时，就以它为条件，进行查询，结果显示在窗体 2 上。窗体 3 作为主窗体。

ListView 和 TreeView 可以通过 HitTest 来实现对各项的控制，那么 ListBox 有没有方法实现对其项的控制呢？答案是肯定的。我们同样可以控制 ListBox 的项，但必须要作以下说明：

```
Private Declare Function SendMessage Lib "user32" Alias "SendMessageA" _
    (ByVal hwnd As Long, ByVal wParam As Long, ByVal lParam As Long, _
    lParam As Any) As Long
Private Const LB_ITEMFROMPOINT = &H1A9
```

制作步骤

一、窗体界面的建立

- 1. 启动 Visual Basic 6.0，新建一个标准工程。
- 2. 向窗体 1 中添加一个 ListBox 控件；向窗体 2 中添加 4 个 CommandButton 控件，其属性中的 Caption 分别设置为 “<<”、“<”、“>”、“>>”，向窗体 2 中添加两个 Label 控件；由于窗体 3 中没有控件，所以没有演示，详细设置请参阅本书光盘源代码。完成界面如图 4-2、图 4-3 所示。

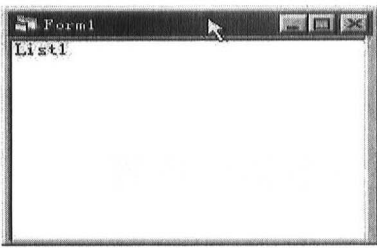


图 4-2 界面图（1）

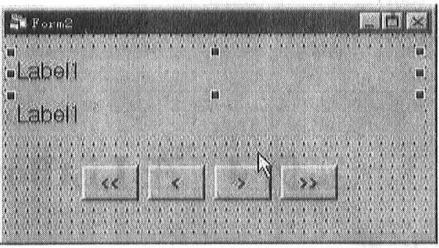


图 4-3 界面图（2）

二、源代码与注释

子窗体显示数据库的代码：

```

Const MeWidth As Integer = 4755
Const MeHeight As Integer = 2655
Private Sub Command1_Click(Index As Integer)
    Select Case Index
        Case 1 '前一条
            rsAuteur.MovePrevious
            If rsAuteur.BOF Then MsgBox "eerste record", vbInformation: rsAuteur.MoveFirst
        Case 2 '下一条
            rsAuteur.MoveNext
            If rsAuteur.EOF Then MsgBox "laatste record", vbInformation: rsAuteur.MoveLast
        Case 0 '第一条
            rsAuteur.MoveFirst
        Case 3 '最后一条
            rsAuteur.MoveLast
    End Select
    Call InvullenGegevens
End Sub

```

```

Private Sub Form_Load()
    Me.Height = MeHeight
    Me.Width = MeWidth
    Call ShowCommand(False)
    Label1(0).Caption = ""
    Label1(1).Caption = ""
End Sub

```

```

Private Sub Form_Resize()
    '最小化时退出
    If Me.WindowState = vbMinimized Then Exit Sub
    '低于标准值则还原
    If Me.Height < MeHeight Then Me.Height = MeHeight
    If Me.Width < MeWidth Then Me.Width = MeWidth
End Sub

```

模块内的代码:

```

Option Explicit
Global Const DBName As String = "biblio.mdb"
Global strPath As String
Global db As Database
Global rsTitles As Recordset

```

```

Global rsAuteur As Recordset
Public Sub GeefDetails(source As String)
    Dim strSQL As String
    Screen.MousePointer = vbHourglass
    strSQL = "SELECT Titles.Title, Titles.ISBN, Authors.Author"
    strSQL = strSQL & " FROM Titles INNER JOIN (Authors INNER JOIN [Title Author] ON
Authors.Au_ID = [Title Author].Au_ID) ON Titles.ISBN = [Title Author].ISBN"    SQL 查询
    strSQL = strSQL & " WHERE (((Titles.Title) = " & source & ")))"
    Set rsAuteur = db.OpenRecordset(strSQL)
    If Not (rsAuteur.BOF And rsAuteur.EOF) Then
        frmMunir2.Caption = source
        rsAuteur.MoveLast
        If rsAuteur.RecordCount > 1 Then Call ShowCommand(True) Else Call ShowCommand(False)
        rsAuteur.MoveFirst
        Call InvullenGegevens
    Else
        frmMunir2.Caption = "niets gevonden"
        frmMunir2.Label1(0).Caption = ""
        frmMunir2.Label1(1).Caption = ""
    End If
    Screen.MousePointer = vbNormal
End Sub

Public Sub InvullenGegevens()
    '在 LABEL 控件里显示值
    frmMunir2.Label1(0).Caption = rsAuteur.Fields(1).Value
    frmMunir2.Label1(1).Caption = rsAuteur.Fields(2).Value
End Sub

Public Sub ShowCommand(zichtbaar As Boolean)
    Dim bX As Byte
    If zichtbaar Then
        For bX = 0 To 3
            frmMunir2.Command1(bX).Visible = True
        Next bX
    Else
    End If
End Sub

```

本实例完整代码请参阅本书光盘。

实例 5 用 ListView 控件显示数据库里的字段

实例效果

程序运行后，单击显示按钮，将会把数据库里的信息添加到 TreeView 里。我们将会看到从“A”到“Z”的 26 个字母。如果该数据库里的字段是以什么字母开头，那么该字段将作为该字母的子树。再单击这个字段，该字段的数据将显示在右侧的 ListView 控件里，如图 5-1、图 5-2 所示。

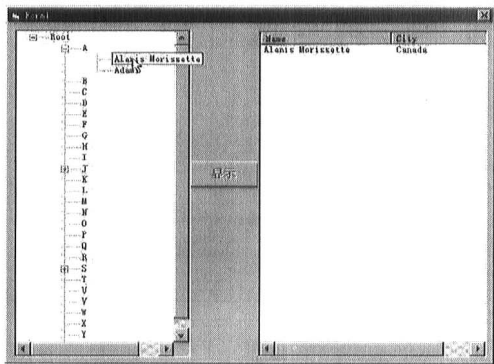


图 5-1 运行界面图（1）

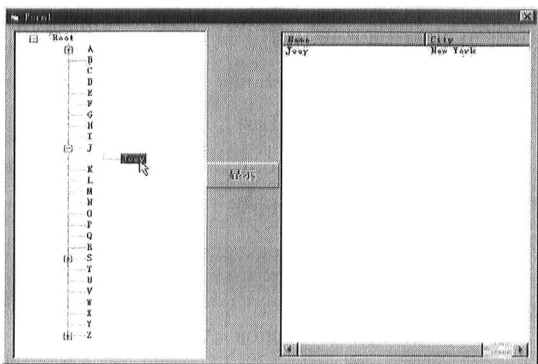


图 5-2 运行界面图（2）

程序说明

本实例主要利用 ListView 控件的 Add 方法添加 ColumnHeader 对象到 ColumnHeaders 集合中。ListView 控件是 Mscomctl.OCX 文件中一组 ActiveX 控件的一部分。若要在应用程序中使用 ListView 控件，则必须将 Mscomctl.OCX 文件添加到工程中。当发布应用程序时，先将 MSCOMCTL.OCX 文件安装到用户的 Microsoft Windows System 或 System32 目录下。

在 VB6 专业版或者企业版中，ListView 控件并不是默认的工具箱的配置项目，必须要手工添加，单击“工程”菜单项目，选择“部件”。在部件对话框的“控件”栏目中选择“Microsoft Windows Common Controls 6.0”，确认在它前面的复选框中有一个黑色的对号，单击“确认”按钮，这时工具箱中便添加了几个控件，其中，有 5 个空心圆圈的便是 ListView 控件了，当然怎么把它添加到窗体上就不再详细介绍了。

作为一个增强功能的列表框，它的本质是用来显示更多的信息。因此，它的 View 属性的 Lvwreport（报表）值就很重要。至于它的外观，读者可以随心所欲地改。

制作步骤

一、窗体界面的建立

1. 启动 Visual Basic 6.0，新建一个标准工程。
2. 向窗体中添加一个 TreeView 控件；向窗体中添加一个 CommandButton，其属性中的 Caption 设置为“显示”；向窗体中添加一个 ListView 控件，详细设置请参阅本书光盘源代码。界面如图 5-3 所示。