


面向对象思想的应用——改进 MSFlexGrid 控件

王振宇 王弘 张春枝 (武汉华中科技大学动力工程系 430074)



摘要: 为了达到更高的功能要求,对 MSFlexGrid 控件进行“升级”。使用面向对象的思想和方法,继承 MSFlexGrid 控件主要原有功能,再以属性和方法的形式设计添加新的功能,并把所有功能(包括附属控件)一并封装到一个新的用户控件(UserControl)或者 ActiveX 控件中。

关键词: 面向对象 MSFlexGrid 控件 用户控件(UserControl) Visual Basic 6.0 SQLServer6.5

Microsoft FlexGrid(简称 MSFlexGrid)控件在 Visual Basic 程序开发、特别是数据库应用中都有着非常广泛的应用。

通常的做法是将其绑定到一个 Data 控件上,那么 MSFlexGrid 显示的将是只读的数据 [1],这样就不能很方便的对数据进行编辑修改,也不能按照人们的习惯直接在网格中新添加一条数据。

为了解决这个问题,目前的策略是将一个 TextBox 控件与 MSFlexGrid 配合使用 [2],在用户点击网格时将 TextBox 嵌入到相应的单元格中,以达到在网格里直接进行文本编辑的目的。

这种做法基本上可以完成功能上的要求,但是,在具体实施中由它所引发的一系列问题却不容忽视。例如,当程序中多处使用 MSFlexGrid 时,代码重复现象就很严重,每使用一个 MSFlexGrid 还都要“搭配”一个 TextBox 和几个 ComboBox(有时为了输入数据方便,在单元格中嵌入下拉框),这就在很大程度上增加了代码维护和管理困难;即使将重复代码写成过程(或函数)也不能很好的解决这个问题,而且过程(或函数)的参数通常比较复杂(因为,一般都要包括一个 MSFlexGrid,一个 TextBox,还有若干个 ComboBox)。

这些问题在网格要完成的功能进一步增多时表现得尤为显著。

所以,最好的解决方案应该是应用面向对象的思想

和方法,在 MSFlexGrid 的基础上,把所有新添加的功能(包括 TextBox 控件和 ComboBox 控件以及在网格中对数据库的操作)一劳永逸地集成到一个新的控件当中,把 MSFlexGrid“升级”。在作者的实际编程中充分体会到了这种方案的优越性。

1 新控件的功能设计

在控件功能方面,由面向对象方法所带来的优点是:新功能的添加和对已有功能实现方法的改变都很容易完成,特别是在不改变控件与应用程序接口的情况下改动量就更小。

尽管如此,在控件的设计初始确定功能比较完备的设计目标还是必要的。初步确定控件的主要功能要求如下:

1.1 基础功能

继承 MSFlexGrid 控件原有的一些常用属性和方法,如确定单元格位置的 Row, Col 等;

嵌入编辑功能如下:

(1) 可以在应用程序中确定哪些列嵌入 TextBox,哪些列嵌入 ComboBox,并可以设置 ComboBox 下拉框的备选项(Item)的内容;

(2) 选择是否允许用户对网格内容进行编辑(可以有三种选择:不可编辑/对全部的列都可以编辑/只对部分列可以编辑);

(3) 使用键盘方便的控制单元格的移动(可以使用上下左右的方向键和回车键 Enter);

(4) 用户鼠标点击网格顶端时,对所点中的列排序(类似 Windows95/98 的资源管理器的功能);

1.2 与数据库相关的功能

将对数据库数据的操作以方法的形式添加到控件中,(以 Visual Basic 与 SQLServer6.5 的数据库连接方式 RDO (Remote Data Objects) 为例) 如对结果集 rdoResultset 对象在网格中的显示、修改、添加、保存、删除等操作功能。

像这样以非绑定方式,手工编制封装在网格控件中的对数据库的操作功能,是一种比较新的尝试。

2 控件的编制

2.1 向新控件添加组成(子)控件

如图 1 所示,新的控件(命名为 Grid)由一个 MSFlexGrid,一个 TextBox 和一个下标为 4 的 ComboBox 数组(此处规定最多只能有 5 个列嵌入下拉框)组成。

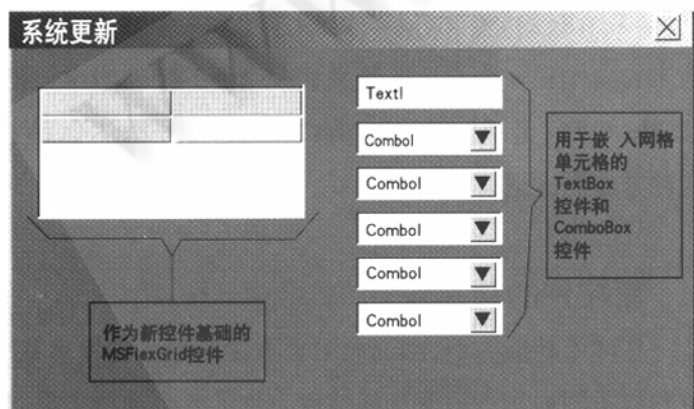


图 1 新控件的组成

2.2 继承 MSFlexGrid 已有的属性和方法

继承 MSFlexGrid 控件已有的属性可以使用 Visual Basic 自带的 ActiveX 控件接口向导帮助完成,也可以手工编写代码。例如下面的 2 段代码就继承了 MSFlexGrid 控件的用于返回或者设置网格列数的属性 Cols。

```
Public Property Get Cols() As Long
'返回网格的列数 Cols
    Cols = MSFlexGrid1.Cols
End Property
Public Property Let Cols(ByVal New_Cols As Long)
'设置网格的列数 Cols
    MSFlexGrid1.Cols() = New_Cols
```

```
PropertyChanged "Cols"
```

```
End Property
```

2.3 添加嵌入编辑功能的属性和方法

为了实现控件封装的严密性,向控件添加属性和方法的机制是这样的:应用程序“看不到”描述控件属性的变量(定义在控件中的私有变量 Private),而只能通过 Public Property Get 和 Public Property Let 这两个过程(函数)返回和设置属性的值,当然,这只是属性的实现机制,应用程序并不知道。

下面一段代码是添加一个新的属性 InlayObjects,标识是否允许用户编辑:

```
Enum EditChoices
```

```
'定义一个枚举变量,用于属性 InlayObjects 值的类型
```

```
[AllColsInlayObject]
```

```
'所有的列都可以编辑,即所有的列都可以嵌入文本框 (TextBox) 等
```

```
[NoColsInlayObject]
```

```
'所有列都不可以编辑
```

```
[PartColsInlayObject]
```

```
'部分列可以编辑
```

```
End Enum
```

```
Dim m_InlayObjects As EditChoices
```

```
'定义新增属性,标记网格是否允许用户编辑(全部列可以,不可以或者部分列可以)
```

```
Public Property Get InlayObjects() As EditChoices
```

```
'返回属性 InlayObjects 的值的函数
```

```
InlayObjects = m_InlayObjects
```

```
End Property
```

```
Public Property Let InlayObjects(ByVal New_InlayObjects As EditChoices)
```

```
'设置属性 InlayObjects 的值的函数
```

```
m_InlayObjects = New_InlayObjects
```

```
PropertyChanged "InlayObjects"
```

```
End Property
```

这样便定义好了属性 InlayObjects,在新的控件收到鼠标或者键盘触发时先检查一下这个属性的值,看是否所在的列允许用户编辑,如果允许(该属性的值不等于 1,即 NoColsInlayObject),再看此列是要嵌入 ComboBox 还是嵌入 TextBox(这就需要再设置一个新的下标为 4 的整数数组类型的属性记录要嵌入 ComboBox 的列号,此数组不包含的列就都嵌入默认的 TextBox)。之后便可以做嵌入操作

了, 嵌入的时候先把 TextBox 设置为不可见, 然后让 TextBox 和网格的有效单元格位置相同大小一样, 再把单元格的内容写到 TextBox 中, 最后把 TextBox 设置为可见、赋予焦点。具体代码如下:

```
With Text1
'嵌入 TextBox
.Visible = False
'下把 TextBox 设置为不可见
.Top = MSFlexGrid1.Top + MSFlexGrid1.CellTop
.Left = MSFlexGrid1.Left + MSFlexGrid1.CellLeft
.Width = MSFlexGrid1.CellWidth
.Height = MSFlexGrid1.CellHeight
'确定 TextBox 的位置和大小
.Text = MSFlexGrid1.Text
.Visible = True
'把 TextBox 设置为可见
.SetFocus
SendKeys "{Home}+{End}"
.ZOrder
End With
```

2.4 添加对数据库中的数据进行操作的功能

作者在程序开发时使用 Remote Data Object (RDO) 与 SQLServer6.5 数据库建立连接, 使用其中的 rdoResultset (或 rdoQuery) 对象对数据进行操作 [3]。网格中对数据进行操作通常是将结果集 rdoResultset 对象和网格严格对应起来, 然后把对网格的操作映射到结果集中。

对数据进行操作通常是显示记录、添加记录、删除记录和修改记录几种。

下面一段示例代码以 Public 过程的形式定义了控件的一个方法 Show, 用于在网格中对结果集数据的显示。

```
Public Sub Show(p_rs_Show As rdoResultset, Optional
p_intColStart As Variant)
Dim intColStart As Integer
'记录从第几列开始显示结果集的内容
If IsMissing(p_intColStart) Then
intColStart = MSFlexGrid.FixedCols
Else
intColStart = p_intColStart
End If
Dim i As Long, j As Long
If p_rs_Show.RowCount > 0 Then
```

```
p_rs_Show.MoveFirst
End If
With MSFlexGrid1
.rows = .FixedRows
i = .FixedRows
Do Until p_rs_Show.EOF
.rows = .rows + 1
For j = intColStart To p_rs_Show.rdoColumns.count +
intColStart - 1
If IsNull(p_rs_Show(j - intColStart)) then
.TextMatrix(i, j)=""
Else
.TextMatrix(i, j) = p_rs_Show(j - intColStart)
End If
Next j
i = i + 1
p_rs_Show.MoveNext
Loop
End With
End Sub
```

3 控件的使用

我们刚刚制作的控件 (UserControl) 可以直接在应用程序中使用, 就像使用其他 Visual Basic 自带的控件一样。他外表类似 MSFlexGrid, 但您会发现使用时不需要一句代码就可以实现在网格中的编辑功能; 而我们定义的一些新的属性和方法可以在代码窗口中由系统提示给出, 如图 2 显示了我们定义的用于显示数据的方法 (Show) 的使用。■

```
Show(P_rs_Show As rdoResultset, [p_intcolStart])
```

图 2 新定义方法的使用

参考文献

- 1 MSDN Library Visual Studio 6.0.
- 2 王若菁, 巧用 Microsoft FlexGrid 控件, 计算机应用, 2000 年 8 月 P70~P72.
- 3 (美) William R. Vaughn, Visual Basic 和 SQL Server 编程指南, 王书洪 陈永基 译。