

一个实用的菜单控件

王瑾 (陕西省咸阳纺织工业学校电教计算机中心 712000)

摘要: 本文介绍了一个简单实用的菜单控件, 可以适用于 C/S 及 B/S 架构, 给应用系统的整合及模块管理带来方便。



1 设计思想

一般的应用系统都有一个主画面, 用户由该主画面进入各个子系统及模块, 这便需要菜单的导引, 就像 windows 的资源管理器一样, 一层一层地打开文件夹, 最后看到各个文件。另外考虑应用系统扩展性 (如增加模块) 及权限控制的要求, 菜单必须是活的 (如菜单结构从数据库中获得)。

2 实现方式

VB 的 TreeView 控件正好适合上述菜单结构的实现, 本文的控件就是由 TreeView 控件及 ImageList 控件组合而成。

对于前面设计的 Menu 表有多种实现方式, 例如采用递归法逐层加入菜单项, 但控制复杂、可能造成死循环、运行时占用系统资源严重。所以本文采用了一种更为简单的方式, 具体是按 Menu_ID 升序排列将菜单表所有记录取出, 然后逐个将菜单项加入 TreeView 控件。由于最高

表 1

Menu表结构						
字段序号	字段英文名	字段中文名	字段类型	允许null	主键字段	字段说明
1	Menu_ID	菜单编号	Char(10)	否	是	每两位代表1级, 最多支持5级。下级菜单编号的前几位即是上级菜单编号, 最高一级菜单编号为零长度字串, 每级的两位编码中第一位为字母A-Z, 第二位为数字0-9, 每级最多包含260个菜单项。
2	Menu_Name	菜单名称	Char(30)	否		对最底层的菜单项记录模块名称, 其他菜单项记录子系统或模块组的名称。
3	Menu_ModuleID	模块编号	Char(30)	否		对最底层的菜单项记录模块编号, 其他菜单项为零长度字串。

关键的核心是一个具有树状层级结构的菜单, 本文以 VB 的 TreeView 控件为基础, 开发出了满足上述要求的实用菜单控件。子系统及模块的层次关系由数据库中的一个表来描述, 具体的表结构如下:

其实 Menu 表也可以设计成其他结构, 例如以菜单编号及上级菜单编号两个字段为主键, 这种表结构虽然支持任意级数的菜单, 但需要维护两个字段的关系容易出错 (例如两个菜单项的编号互为父子关系, 造成死循环)。

因此, 本文采用上述表结构。对于要求层数更多的菜单结构, 可以增加 Menu_ID 字段的长度 (例如 16 即可支持 8 级菜单结构)。

一级菜单项的 Menu_ID 为零长度子串, 该项必然排在第一条, 父菜单项也会排在子菜单项的前面, 所以产生菜单结构时肯定是最先增加根节点, 然后增加第一个一级节点, 再增加第一个属于该节点的二级节点, 直至最底层的模块。这正好复合 TreeView 控件上级节点必须存在才能增加下级节点的特点。

具体设计是该控件包括一个方法 ShowMenu (aMenuInfo), 此方法包含一个二维数组参数 (传递按 Menu_ID 排列的菜单项信息), aMenuInfo(I,0) 表示菜单编号, aMenuInfo(I,1) 表示菜单名称, aMenuInfo(I,2) 表示模

块编号。

该控件还包括一个事件 MyClick(sModuleID As String), 当点击节点时触发此事件, 同时将节点对应的模块编号作为参数传递给此事件, 节点对应的模块编号在调用 ShowMenu 方法时被记录到每个节点的 Tag 属性中。

控件中有一个 ImageList 含有 3 个图片(Open.bmp、Closed.bmp、Module.bmp), 分别对应打开、关闭的菜单项及底层的模块。

3 源代码

图 1、图 2 表示了组成菜单控件的两个控件的属性设置, 后面是完整的源代码。

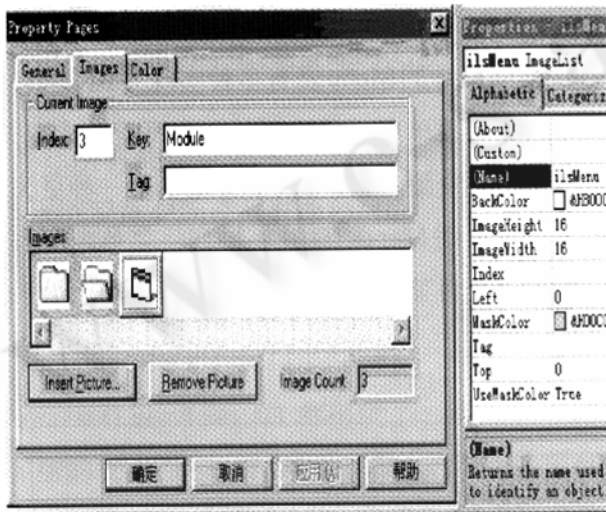


图 1 ImageList 控件

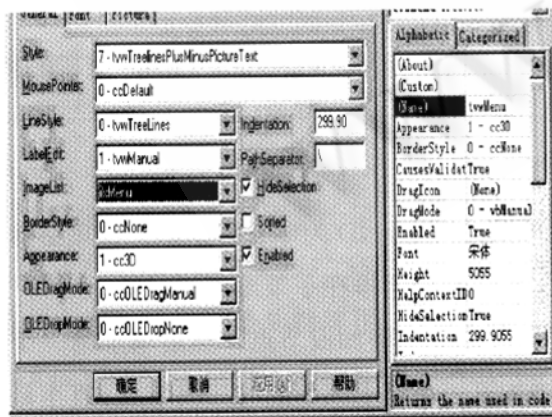


图 2 TreeView 控件

Option Explicit

'声明事件, 该事件与 TreeView 控件的 NodeClick 事件关联,

'同时将被点击模块的编号传递出去, 以便调用相应

的模块。

```
Event MyClick(ByVal sModuleID As String)
```

```
Public Function ShowMenu(aMenuInfo) As Boolean
```

```
Dim i As Long
```

```
Dim sMenuID As String
```

```
Dim sParentID As String
```

```
Dim sMenuName As String
```

```
Dim sModuleID As String
```

```
On Error GoTo errHandle
```

```
For i = 0 To UBound(aMenuInfo)
```

```
'取得菜单编号、菜单名称及模块编号
```

```
sMenuID = aMenuInfo(i, 0)
```

```
sMenuName = aMenuInfo(i, 1)
```

```
sModuleID = aMenuInfo(i, 2)
```

```
If sMenuID = "" Then '加入根节点
```

```
twvMenu.Nodes.Add , , "Root", sMenuName,
```

```
"Closed"
```

```
Else '加入其他节点
```

```
sParentID = Left(sMenuID, Len(sMenuID) - 2)
```

```
If sParentID = "" Then
```

```
sParentID = "Root"
```

```
End If
```

```
If sModuleID = "" Then '非底层节点
```

```
twvMenu.Nodes.Add sParentID, twvChild,
```

```
sMenuID, sMenuName, "Closed"
```

```
Else '底层节点 (即各个模块)
```

```
twvMenu.Nodes.Add sParentID, twvChild,
```

```
sMenuID, sMenuName, "Module"
```

```
'将最底层模块的编号记录到节点的 Tag 属
```

性中

```
twvMenu.Nodes(sMenuID).Tag = sModuleID
```

```
End If
```

```
End If
```

```
Next i
```

```
twvMenu.Nodes("Root").Expanded = True
```

```
ShowMenu = True
```

```
Exit Function
```

```
errHandle:
```

```
MsgBox "显示菜单项失败!" & Chr(13) & Chr
```

```

(10) & Err.Description
    Err.Clear
End Function

Private Sub tvwMenu_NodeClick(ByVal Node As
Node)
    If Trim(Node.Tag) <> "" Then
        RaiseEvent MyClick(Node.Tag)
    End If
End Sub

Private Sub tvwMenu_Collapse(ByVal Node As
ComctlLib.Node)
    Node.Image = "Closed"
End Sub

Private Sub tvwMenu_Expand(ByVal Node As
ComctlLib.Node)
    Node.ExpandedImage = "Open"
End Sub

Private Sub UserControl_Resize()
    '使菜单显示区域与控件大小一致
    tvwMenu.Width = UserControl.Width
    tvwMenu.Height = UserControl.Height
End Sub

```

4 应用示例

4.1 C/S 架构

该应用实例采用一个MDIForm作为主画面，主画面左侧放置上述的菜单控件，右侧是一个MDIChild窗体，负责显示各个模块。每个模块实际上就是一个大控件

(.ctl)，这种开发方式相当于将每一模块独立调试、封装，最后一并整合，可以避免大规模开发时变量命名冲突、难于整合等问题。具体的工程文件信息如图3所示。

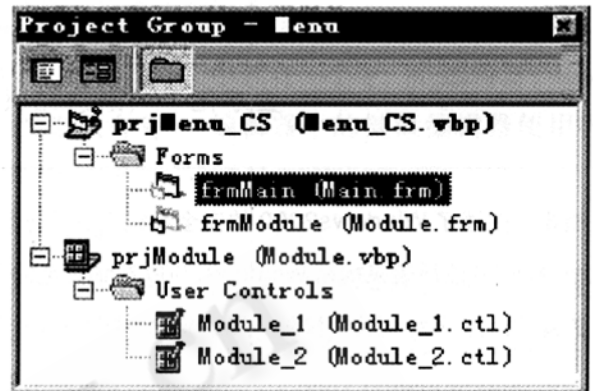


图3 工程文件信息

4.2 B/S 架构

该应用实例仍然由一个主画面 (Main.htm) 开始，其中包含左右两帧，左帧 (Manu.asp) 容纳菜单控件，右帧 (fraModule) 容纳各个模块，核心部分在左帧。

显然菜单控件需要从数据库获取层级结构信息，控件又要下载到客户端运行，而B/S架构一般不允许在客户端设置数据源，因此为了取得层级结构信息必须把数据传递到一个客户端脚本中的数组，然后在页面初始化时显示菜单。

具体实现的源代码略。■

