ASP. NET 中 TreeView 控件的动态绑定及定位展开

Dynamic Binding and Positioning about TreeView Control under ASP. NET

浮光宾 王葵如 张明伦 （北京邮电大学 光通信与光波技术教育部重点实验室 北京 100876）

摘 要: TreeView 控件是 web 开发中常用到且功能强大的控件。本文主要阐述在 Visual Studio2005 环境开发 B/S 系统中树形控件的使用方法及其技巧，重点介绍一种基于 MSSQL 数据库树形控件的动态绑定及参数回传定位展开树形结构的方法。

关键词: 树形结构 节点 TreeView B/S 系统 动态绑定

树形图用于显示进行组织树形结构的数据，其使用比较广泛，如计算机的资源管理器、公司组织结构等[1]。在 Web 开发中常利用独立的树形控件进行树形图的开发，其在页面常充当导航器的作用。而在实际使用中，很多情况下需要将树形控件与数据库进行连接，进行动态绑定显示节点及增删节点。本文中所举的例子是在 VS2005 IDE 环境中用 VB 语言进行开发的，VS20005 已经集成了 TreeView 树形控件，使用起来非常的方便，不用像 VS2003 需另行安装。下面将以 MIB 的树形结构图为例介绍 TreeView 控件的动态绑定及参数回传定位展开树形结构的方法。

1 TreeView 控件的一些常用属性和方法

TreeView 控件由节点组成，树中的每一项都称为一个节点（TreeNode）。TreeView 控件的属性、方法及事件有多种，可以参看 MSDN 帮助文档，本文只列举一些常用的及本例中所用的属性、方法及事件[2]。如表 1 表 2 所示：

表 1 TreeView 的一些常用属性、方法和事件

方法/事件	说明
Nodes	获取 TreeNode 对象的集合，它表示 TreeView 控件中根节点
SelectedValue	获取选定节点的值
ExpandAll	打开树中的每个节点
OnSelectedNode Changed	引发 SelectedNodeChanged（当折叠 TreeView 控件中的节点时发生）事件
OnTreeNodeExp anded	引发 TreeNodeExpanded（当扩展 Tree-View 控件中的节点时发生）事件
OnTreeNodePopu-late	引发 TreeNodePopulate（TreeView 控件中展开时发生）事件

表 2 TreeNode 的一些常用属性

属性	说明
ChildNodes	获取 TreeNodeCollection 集合，该集合包含当前节点的第一级子节点
PopulateOn-Demand	获取或设置一个值，该值指示是否动态填充节点
Text	获取或设置为 TreeView 控件中的节点显示的文本。
Value	获取或设置用于存储有关节点的任何其他数据（如回发事件的数据）的非显示值 其中，TreeView 的属性 Nodes，TreeNode 的属性 ChildNodes 都可以通过 Add，de-lete 等方法进行节点的添加删除修改等。

2 数据库表结构设计

考虑到 MIB 的结构，我们暂时先建立两个表，表 treenode 中存放树节点的信息，表 leafnode 里面存放叶节点的信息。如表 3 所示：

表 3　treenode/leafnode 表

字段名	数据类型	描述
ID	Integer	节点 ID(自增)(主键)
ParentID	Interger	父节点 ID(根节点的此值定为 0)
Depth	Interger	节点深度(根节点深度为 0)
OID	Interger	OBJECT IDENTIFIER
Name	Text	对象名称
……	……	……

上表列出的为 Treenode 表和 leafnode 表相同部分的字段,……为不同的地方,根据具体的要求进行定义。接下来就可以在 MS SQL 里面创建数据库 MIB.mdb 并创建本例中所用到的两个表 treenode 和 leafnode[3]。

3　树形控件的使用及其与数据库的绑定连接

数据库建好以后就可以在 VS2005 IDE 中创建树形结构页面,首先添加页面 MibTree.aspx,接着从工具框中拖放树形控件到页面即完成了树形控件的添加,剩下的工作就是对其节点数据的添加和绑定及控件事件、效果的添加和设定,这个本文的重点部分。

3.1 数据库的连接

因为本系统的建设考虑到其扩展及代码可重用性,按照三层(多层)结构进行创建,所以数据层单独列出并做为一个类进行创建。如 DBAcess.vb 类,图 1 列出了此类提供的一些接口和函数:

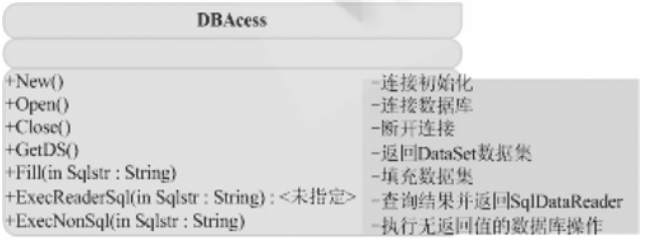

图 1　DBAcess 类图

接下来就可以通过定义 DBAcess 的对象而实现其与数据库的连接,如下定义了一个函数,通过与数据库的连接,取出数据并保存在 DataSet 数据集中返回。

```
Function RunQuery ( ByVal sqlstr As String ) As DataSet
    Dim DBA As DBAccess
    Dim Ds As DataSet
    DBA = New DBAccess()
    DBA. Fill(sqlstr)
    DBA. Close()
    Ds = New DataSet()
    Ds = DBA. GetDS()
    Return Ds
End Function
```

下面与数据库的交互将通过调用此函数进行数据的提取,本例子中不涉及树形结构的添加,所以就不讲 ExecNonSql 函数的使用。

3.2 树形控件的动态绑定

对树形控件的绑定是从根节点开始的。我们对节点进行动态添加,根节点的深度定义为 0,以下节点每层依次增 1。

根节点是页面加载前绑定添加的,如下:

在 Page_Load 下面添加方法 PopulateRoot(),动态添加根节点,

```
Public Sub PopulateRoot()
Dim Ds As DataSet = RunQuery("根结点执行语句")
    If Ds. Tables. Count > 0 Then
      Dim row As DataRow
      For Each row In Ds. Tables(0). Rows
       Dim newNode As TreeNode = New TreeNode()
            newNode. Text = row ( " Name " ). ToString. Trim
             newNode. Value = row ( " ID " ). ToString. Trim
          newNode. PopulateOnDemand = True
          TreeView1. Nodes. Add(newNode)
        TreeView1 为树形控件 ID,添加根节点
      Next
    End If
```

End Sub

接下来添加树形控件的 TreeNodePopulate(TreeView 控件中展开时发生)事件,当点击节点时,如果有下一层节点时,将动态的绑定展开,在事件下面添加方法

Dim Depth = e.Node.Depth

PopulateSubnode(e.Node,Depth)' 绑定节点函数

其中 Node 为选中节点,Depth 为结点深度

PopulateSubnode 函数的具体实现过程基本上同 PopulateRoot 函数:不同的是 RunQuery(sql)中的 sql 执行语句,此语句应包含对叶结点表的查询,绑定过程中分两部分判断,分别对辅助子结点及叶结点的添加。

函数 PopulateRoot 和 PopulateSubnode 均采用了根据绑定范围和条件,从数据库中取出相关的数据,并轮询动态添加在其父节点下或根节点位置。其设计流程图如图 2 所示:

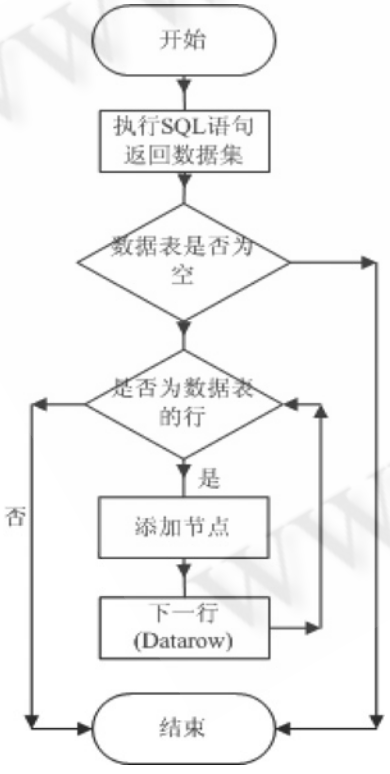

图 2　树形绑定(展开)流程图

3.3 参数回传树形控件节点的定位展开

本功能的实现是利用节点的遍历来实现的,上面没有涉及到节点的添加修改删除,实际节点的这些执行过程是在另一些页面中操作的。要让树形结构根据这些操作进行实时的更新则需要进行参数的回传,根据要求来刷新树结构。本例定义了此实现方法(ExpandDepth(ID,model)),有两个传递变量,分别为节点的 ID 和类型(辅助节点或叶结点),此方法在页面加载时执行。实现函数体的主要实现代码如下:

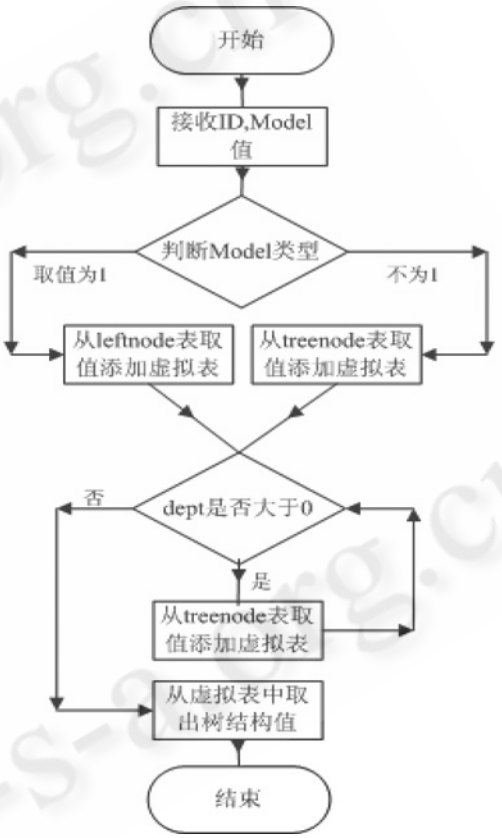

图 3　MS SQL 树结构循环提取流程图

```
……
Dim Ds As DataSet = RunQuery(" Sqlstf" )
  If Ds.Tables.Count > 0 Then
    Dim row As DataRow
      Dim FNode As TreeNode = New TreeNode()
       For Each row In Ds.Tables(0).Rows
         If row(" depth" ) =0
         Dim node As TreeNode = New TreeNode
         For Each node In TreeView1.Nodes
           If row(" nodeid" ) = node.Value Then
             node.Expanded = True
```

```
            FNode = node
            Exit For
          End If
        Next
      Else
          FNode = expd(FNode,row(" nodeid" ))
        End If
      Next
    End if
End Sub
```

上面函数采用了的分层节点轮询查找展开,其中 Sqlstr **使用下面语句定义,采用了对树形结构数据表遍历查询的算法进行提取(如图** 3 **所示)**[4]**。程序需要临时存储记录集,在** SQL **语句中用临时表来实现,执行查询功能最有效快速的是使用存储过程,其设计循环提取树形数据算法流程图如图** 3 **所示。不过本例子只是列出了所使用的主要** SQL **语句,如果使用存储过程可以对其进行创建和调用。**

```
declare @ id as integer
set @ id = " 传递的 ID 值"
declare @ model as integer
set @ model = " 传递的节点类型" 1 为叶节点
declare @ dept as integer
declare @ tnode table ( ID integer IDENTITY ( 1,
1) not
null,nodeID integer, depth integer )
if @ model < >1
   begin insert @ tnode(nodeID, depth) select
   parentid,depth -1 from treenode where id = @
id end
  else
   begin insert @ tnode(nodeID, depth) select
   parentid,depth -1 from leafnode where id = @
id end
select @ id = nodeID, @ dept = depth from
@ tnode
where id = @ @ identity
while @ dept >0
   begin
      insert @ tnode(nodeID,depth) select parentid,
depth -1
      from treenode where id = @ id
       select @ id = nodeID, @ dept = depth from
@ tnode
      where id = @ @ identity
   end
 select * from @ tnode order by id desc
```

下面函数 expd(pnode,ID) **是轮询查找展开对应一个节点的方法,供** ExpandDepth **进行调用,传递参数是分别上一级节点和下级节点的** ID**,返回类型为** TreeNode**。函数体实现如下:**

```
Dim node As TreeNode = New TreeNode
  For Each node In pnode. ChildNodes
    If ID = node. Value Then
      node. Expanded = True
      Exit For
    End If
  Next
Return node
```

4 总结

树形控件的动态绑定和树结构表的遍历查询是涉及到与数据库交互的编程中经常遇到的问题。本文通过 MIB **树形结构的实现过程来介绍了树形控件的一些常用的使用方法和一些使用的技巧。树形控件是一个常用的控件,可以根据需求采取科学的使用方法,开发出高效且功能丰富实用的程序。**

参考文献

1 王延红,杨平利,黄少华. Visual C + +. NET 中树形控件的使用. 现代电子技术,2006,(14):61 -63.

2 奚江华. 圣殿祭司的 ASP. NET 2.0 开发详解. 电子工业出版社: 北京,2006,(11).

3 汪华斌. TreeView 在主从表关系处理中的应用. 计算机与现代化,2006,(3):74 -75.

4 吉杰,陶培基. MS SQL Server 树形结构表遍历的循环算法. 计算机与现代化,2005,(4):7 -8.