

动态 WEB 数据库的多级关联下拉框选项的设计技术

刘克胜 冯涛 郭瑞萍 (兰州 27 支局 11 信箱 67 号 732750)

摘要: 本文主要描述了 Microsoft Visual Interdev 6 的开发动态 WEB 的优点, 利用它设计动态 WEB 数据库的多级关联下拉框, 对一般动态 WEB 数据库网页的设计也提供了指导方法。

关键词: ASP 设计期间控件 数据库 脚本对象模型 属性

本文主要介绍运用 Visual InterDev6 开发动态 WEB 数据库的多级关联下拉框选项。

1 选择 Visual InterDev 的优点

Microsoft Visual InterDev 提供快速的开发环境、功能强大的数据库工具以及集成化站管理功能, 可使开发人员轻松地创建、出版并管理新一代动态 Web 应用。它是为程序员完成下列任务而设计的 Web 开发工具:

(1) 数据驱动 Web 应用: 使用由 ODBC 或 OLE DB 所支持的数据资源, 例如得到微软公司数据库管理系统支持的数据资源。

(2) 使用 HTML 和含脚本的广泛 Web 页面: 在这样页面中使用最先进的浏览器技术, 例如使用 Internet Explorer 5.0、动态 HTML 和多媒体特性。

(3) 强健的开发环境: 使用脚本编程模型、设计期间控件 (DTC) 和一种扩展的工具箱。这个工具箱用于页面的快速设计, 并进行测试和调试。

(4) Web 工作组: 他们可以独立开发页面, 并随时准备访问和维护主版本; 也可以利用非程序人员用 Microsoft FrontPage 在主版本上进行工作。

(5) 综合解决方案: 其中包括一些小应用程序或部件, 它们是用 Microsoft Visual Basic、Visual C++、Visual J++ 等工具软件创建的。

Microsoft Visual InterDev 还提供了一个傲视群雄的脚本对象模型, 它定义一组对象、事件、属性和方法, 你可以使用它们建立你的应用, 并为这些应用编写脚本。你可以使用设计期间控件, 为你的应用建立 Visual 接口; 然

后编写脚本, 使用传统的面向对象的技术控制应用。如果你把脚本对象模型同如何使用传统 HTML 方法创建 Web 应用比较起来, 这种脚本模型是最容易理解的。要在传统 HTML 中创建一种窗体, 例如, 你要把一些内部 HTML 控件放到一个页面上, 其中包括文本框、列表框和按钮。要使用传统 HTML 方法为这些控件编写脚本, 你就不能简单地编写事件处理程序或调用这些控件的方法。你要依靠浏览器把这种窗体提交给服务器上的页面。这个目的页面必须包含对浏览器发送的串进行分析的脚本。相比之下, 这种脚本对象模型允许你用控件, 也允许你使用标准面向对象技术的页面进行工作。例如, 不使用 HTML 要求的复杂的窗体提交过程, 你可以简单地把按钮放到页面上, 并为它的 onclick 方法编写一个处理程序, 处理这个窗体。脚本对象模型提供下述好处:

① 使用熟悉的面向对象模式迅速开发应用。你可以把标准面向对象技术用于开发 Web 应用。因为你只有编写脚本量少, 脚本使用又能很简单, 你才能迅速创建应用, 并且错误又少。

② 浏览器和平台独立。你可以使用脚本对象模型, 不管什么浏览器将访问你的应用。不管你为服务器脚本 (为最大距离) 还是为客户脚本 (为性能) 设计你的应用, 实际上工作都是一样的。

③ 简化窗体。你可以使用把设计期间控件拖到一个页面上的办法创建窗体, 其方式同你在诸如 Visual Basic 环境中使用的方式相同; 再使用属性改变它们的样子和表现。你可以通过编写标准事件处理程序被调用方法的办法处理窗体, 这个脚本对象模型完成复杂的寄送窗体的任务

并分派给正确的窗体处理逻辑。

④孤立的应用逻辑。你可以很容易地在分立的过程中把你的应用孤立出来,其中包括其他页面上的过程,不是把它同用户接口和导航元素混为一谈。

⑤状态维护。脚本对象模型提供在各页面之间共享信息的机制,无需手动往返于隐蔽段和查询串之间。

⑥数据绑定。脚本对象可以绑定到数据段上,从而你可以在数据录入和数据编辑窗体中使用这些数据。



图 2 DTC 的 Listbox 控件

2 动态 WEB 数据库多级关联下拉框设计

首先,我们运用 Microsoft Access 建立一个名为 multi_level.mdb 的数据库,在数据库中建立三个表,表 1 名为 level1,它包含字段为 level1_id、level1_name;表 2 名为 level2,它包含字段为 level1_id、level2_id、level2_name;表 3 名为 level3,它包含字段为 level1_id、level2_id、level3_id、level3_name。建立表后往数据库中输入内容,如表 1 中地球上各大洲及其编号,表 2 中输入各大洲编号所包含的国家及其编号,同样在表 3 中输入各国家所含的省会。然后在控制面板的 ODBC 数据源中建立系统 DSN 数据源名为 multi_select。

现在,我们在 Microsoft Visual Interdev 中建立一个新的 WEB 工程名为 project1。在工程中增加一个数据连接,右击 project1 在弹出的菜单中选中 Add Data Connect 项,选择系统 DSN 数据源名为 multi_select 的连接,然后加入一空白 WEB 页 multi_select.asp。由于篇幅所限,图表就不再给出。下面是该页具体设计的主要步骤:

(1) 加入三个 DTC 的 Recordset 控件 Recordset1、Recordset2、Recordset3, Recordset1 选择表 level1 且以自动方式打开, Recordset2 选择 SQL 语言“select * from level2”且取消自动打开, Recordset3 选择 SQL 语言“select * from level3”且取消自动打开(见图 1)。

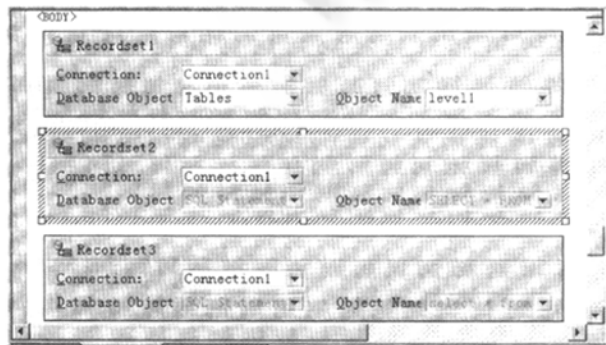


图 1 DTC 的 RecordSet 控件

(2) 加入三个 DTC 的 Listbox 控件 Listbox1、Listbox2、Listbox3 它们分别绑定 Recordset1、Recordset2、Recordset3。Bound Column 绑定为相应的 ID 项, List field 绑定为相应的 name 项(见图 2)。

(3) 加入一个 DTC 的 PageObject 控件,再其中加入三个 SESSION 变量 snlv1_id、snlv2_id、snlv3_id(见图 3)。

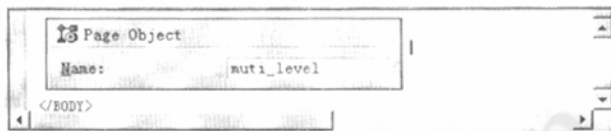


图 3 DTC 的 PageObject 控件

(4) 利用 Script Outline 加入如下函数:

```
Sub Recordset1_onrowenter() // 记录 Recordset1 进入事件, 得到一级选择 ID 号
```

```
    thisPage.setsnlv1_id (Listbox1.getValue  
(Listbox1.selectedIndex))
```

```
    dim temp_id
```

```
    temp_id = thisPage.getsnlv1_id ()
```

```
    if temp_id = "" then thisPage.setsnlv1_id ("01")
```

```
    recordset1.close ()
```

```
End Sub
```

```
Sub Listbox1_onchange() // 控制下级下拉框的显示
```

```
    dim temp1_id,temp2_id
```

```
    temp1_id = thisPage.getsnlv1_id () //取得本级下拉框  
    的 ID 号
```

```
    listbox2.clear () //清空前一次二级下拉框的内容, 为  
    下一次下拉框显示作好准备
```

```
    if recordset2.isOpen() then recordset2.close end if
```

```
    sql2 = "select * from level2 where level1_id =  
    "&temp1_id&"// 依一级下框的
```

Recordset2.setSQLText sql2 // 内容选择数据库的二级下拉框表的内容

recordset2.open // 打开数据库为下拉框提供数据源

if recordset2.EOF then

recordset2.close

thisPage.setsnlv2_id ("00")

thisPage.setsnlv3_id ("00")

Response.Write (" 二级选项为空 ")

end if

if thisPage.getsnlv2_id () <> "00" then // 二级下拉框数据源非空的情况

thisPage.setsnlv2_id (listbox2.getValue (listbox2.selectedIndex))

temp2_id = thisPage.getsnlv2_id ()

if temp2_id = "" then thisPage.setsnlv2_id ("01")

temp2_id = thisPage.getsnlv2_id ()

listbox3.clear () // 清空前一次三级下拉框的内容

if recordset3.isOpen() then recordset3.close end if

sql3 = "select * from level3 where level1_id = "&""&temp1_id&""&" and

"&"level2_id="&""&temp2_id&""

Recordset3.setSQLText sql3

recordset3.open // 打开数据库为下拉框提供数据源

thisPage.setsnlv3_id (listbox3.getValue (listbox3.selectedIndex))

if recordset3.EOF then

recordset3.close

thisPage.setsnlv3_id ("00")

Response.Write (" 三级选项为空 ")

end if

end if

End Sub

Sub Recordset2_onrowenter() // 记录Recordset2进入事件, 得到二级选择 ID 号

thisPage.setsnlv2_id (Listbox2.getValue (Listbox2.selectedIndex))

dim temp_id

temp_id = thisPage.getsnlv2_id ()

if temp_id = "" then thisPage.setsnlv2_id ("01")

End Sub

Sub Listbox2_onchange() // 控制下级下拉框的显示

Dim temp1_id,temp2_id

temp1_id = thisPage.getsnlv1_id ()

temp2_id = thisPage.getsnlv2_id ()

listbox3.clear () // 清空前一次三级下拉框的内容

if recordset3.isOpen() then recordset3.close end if

sql3 = "select * from level3 where level1_id = "&""&temp1_id&""&" and

"&"level2_id="&""&temp2_id&""

Recordset3.setSQLText sql3

recordset3.open // 打开数据库为下拉框提供数据源

if recordset3.EOF then

recordset3.close

thisPage.setsnlv3_id ("00")

Response.Write (" 三级选项为空 ")

end if

End Sub

Sub Recordset3_onrowenter() // 记录 Recordset3 进入事件, 得到三级选择 ID 号

thisPage.setsnlv3_id (Listbox3.getValue (Listbox3.selectedIndex))

dim temp_id

temp_id = thisPage.getsnlv3_id ()

if temp_id = "" then thisPage.setsnlv3_id ("01")

End Sub

Sub Listbox3_onchange()

dim temp1_id,temp2_id,temp3_id

// 得到下拉框选项的内容, 以方便其他应用的使用

temp1_id = thisPage.getsnlv1_id ()

temp2_id = thisPage.getsnlv2_id ()

temp3_id = thisPage.getsnlv3_id ()

if recordset3.isOpen() then recordset3.close end if

End Sub

记录集的 onrowenter 事件是当游标移向 Recordset 脚本对象中的一个记录时, 将触发 onrowenter 事件, 本工程中主要用它获得前一级和当前一级下拉框的 ID 编号。onchange 事件是对 Listbox 脚本对象中的内容作更改或设置时, 将触发 onchange 事件。请注意, 为了触发该事件, 可能要移动焦点。本工程中主要用它控制本级和下一级下拉框的内容的显示和更新。至此大功告成。

现在你可以加入别的内容, 以达到美化你的页面和增加新的功能的目的。■