

基于 ASP.NET 的 WEB 数据库的分页显示

Pagination Display of WEB DataBase Based on ASP.NET

胡业江 张全伙 (福建泉州国立华侨大学信息科学与工程学院 362011)

摘要: ASP.NET 是一种服务器端制作动态 WEB 页的全新编程手段。本文讨论了在 ASP.NET 框架下对 WEB 数据库实现分页显示的方法, 并给出了一个利用该方法实现分页的实例。

关键词: ASP.NET WEB 数据库 DataGrid 分页

随着计算机网络技术的发展和普遍应用, 人们的工作和生活越来越离不开网络, 在网上查询一些信息已是司空见惯的事情。有时候, 如果查询的信息比较多, 结果密密麻麻地显示在浏览器窗口, 看上去很不舒服。另外, 如果一屏显示不下, 还要借助于窗口滚动条, 显然也很麻烦。但是, 若对查询的结果分页显示, 效果将截然不同。从用户的角度, 这种分页显示的结果看起来整齐简洁; 从系统的角度, 分页显示的速度比一下子显示整个查询记录来得快, 因而可以节省网络带宽, 增加系统的反应速度。本文讨论了在 ASP.NET 框架下实现查询记录分页显示的技术。

1 ASP.NET 简介

ASP.NET 是一种创建动态 WEB 应用程序的全新手段, 它建立在 Microsoft .NET 架构平台上, 利用“普通语言运行时库”(Common Language Runtime)在服务器端为用户提供建立强大的企业级 WEB 应用服务的编程框架, 它

与 ASP3.0 完全兼容, 即只要把 ASP 程序的扩展名 .asp 改为 .aspx 就可以在 .NET 框架下运行。ASP.NET 主要包括 WebForm 和 WebService 两种编程模式。前者为用户提供建立功能强大、外观丰富的基于表单(Form)的可编程 Web 页面; 后者通过对 HTTP、XML、SOAP、WSDL 等 Internet 标准的支持提供在异构网络环境下获取远程服务、连接远程设备、交互远程应用的编程界面。

2 ASP.NET 连接数并操纵数据库的方法和步骤

在 Microsoft 公司推出开放式数据库互连 (ODBC) 应用程序编程接口 (API) 以来, 出现了各种各样的数据库访问技术, 而 ADO.NET 是至今为止其中最新的一种数据库接口编程技术。ADO.NET 比它前面的版本 ADO 更遵循数据库访问的通用原则, 面向普通数据库。通过 ADO.NET 不仅能够使应用程序连接到数据源, 检索、操纵和更新数据, 而且能够使应用程序实现对非关系型数据库格式数据的访问。

ADO.NET 提供了一套完整的管理数据访问的集合, 其中有 3 个基本对象是经常使用的。它们分别是: Connection(连接)、Command(命令)、DataSet(数据集)。

2.1 用 Connection 对象连接到数据源

连接到不同的数据源 (SQL Server 数据库、Microsoft Access 数据库、ODBC 数据源等) 有不同的方法。在后面的例子中, 本文采用的是连接到 ODBC 数据源, 因为通过 ODBC 可以访问更多类型的数据库。在配置了服务器的 ODBC 数据源之后, 就可以用以下语句创建一个与它相连接的对象:

```
Dim ConnectionStr As String
Dim MyConnection As ADOConnection
ConnectionStr="FileDSN=数据源文件名"
MyConnection=New ADOConnection
(ConnectionStr)
```

2.2 用 Command 对象操纵数据源

在创建了对数据源的连接后, 需要创建 Command 对象以实现对数据库的操纵。常用的返回记录集的对象有 DataReader 和 DataSet

两种,但由于前者提供的是一个只读和只转发的、单向向前移动的记录集,而后者能自由在记录集中前后移动,并能容纳多个记录集及包括各个集之间的关系,根据分页显示的特性,本文选用后者。语句如下:

```
Dim SQLStr As String
Dim MyCommand As
ADODataSetCommand
Dim DS As DataSet
SQLStr="select * from 表名"
MyCommand = New
ADODataSetCommand (SQLStr, MyConnection)
DS=New DataSet()
MyCommand.FillDataSet(DS, "表名")
```

2.3 用 DataGrid 控件分页显示记录

在创建连接和命令对象之后,就要把返回的记录显示出来。DataGrid控件是ASP.NET服务器的数据列表控件中的一种类型,它的作用是在页面上以表格的形式显示数据。它具有自动创建包含所需行数的页的逻辑,而且可以以多种方式提供页面导航功能。当使用DataGrid控件显示DataSet对象存储的数据记录时,需要将DataGrid控件与数据源绑定,绑定的实质就是确定数据源。而要实现这种绑定,则必须先创建DataView对象(比如是Dview,创建的方法在后面介绍),然后由它来完成,因为DataView对象能显示出整张表或者表行中的子集。语句如下:

```
MyDataGrid.DataSource=Dview
MyDataGrid是DataGrid控件的一个实例
MyDataGrid.DataBind()
```

DataGrid控件控制页面导航的属性和方法见表1,显示风格的属性见表2。由表1可以看出,要启动分页功能,必需将AllowPaging属性设置为True,并指定在发生PageIndexChanged事件时将要运行的事件处理程序的名称。只有把Mode属性设置为NextPrev时,NextPageText和PrevPageText两属性才会起作用。

表1 DataGrid 控件控制页面导航的属性和方法

属性或方法	描述
AllowPaging	用于表明是否允许对DataGrid控件的内容进行分页,取值为Boolean型,缺省值为False
CurrentPageIndex	用于获取或指定当前显示页的索引值(序号)
PageCount	用于返回分页后的总页数,只读属性
PageSize	用于获取或指定在每一页上显示的记录数量,缺省值为10
PagerStyle	用于指定页面导航标志的风格,只读属性
OnPageIndexChanged	在不同页之间进行导航时,要产生PageIndexChanged事件,OnPageIndexChanged方法用于为这个事件指定事件处理函数

表2 PagerStyle 的显示风格属性

属性	描述
PagerStyle_Mode	指定页面导航标志使用的模式,可取值有:NextPrev和NumericPages,缺省值是NextPrev
PagerStyle_NextPageText	指定导航到下一页的按钮所使用的文本
PagerStyle_PageButtonCount	当PagerStyle_Mode属性的取值为NumericPages时,指定在导航标志中显示的页的数量
PagerStyle_Position	指定页面导航标志相对于其关联控件的位置,可取值有:Bottom、Top和TopAndBottom,缺省值是Bottom
PagerStyle_PrevPageText	指定导航到上一页的按钮所使用的文本
PagerStyle_Visible	指定页面导航标志是否可见,取值为Boolean型,缺省值为True

3 实例

下面通过一个简单的例子来说明如何使用以上对象和控件以实现WEB数据库的分页显示。该例使用的数据库名称是“ExampleDSN”,它连接的数据库是“StudentInfo.mdb”,该数据库中有一张“Course”表。

首先将创建DataView对象的代码建立一个专门的“CreateDataView.inc”文件,然后将其包含到主程序中。CreateDataView.inc的代码清单如下:

```
<script language="VB" runat="Server">
Function CreateDataView() As DataView
Dim ConnectionStr, SQLStr As String
Dim MyCommand As ADODataSetCommand
Dim DS as DataSet
ConnectionStr="FileDSN=ExampleDSN"
SQLStr="Select * from Course"
MyCommand=New ADODataSetCommand(SQLStr, ConnectionStr)
DS=New DataSet()
MyCommand.FillDataSet(DS,"Course")
'创建了一个与Course表相关联的DataView对象
CreateDataView=New
Data View(DS.Tables("Course"))
```

```

End Function
</script>
    以下是主程序的主要代码清单:
    <% @ Import Namespace="System.Data"
%>
    <% @ Import Namespace="System.Data.
ADO" %>
    <!-- #include file=CreateDataView.inc -->
'包含前面创建的CreateDataView.inc文件
    <html><head>
    <script language="VB" runat="Server">
    Sub Page_Load(Src As Object, E As
EventArgs)
    '每次加载页面时,执行Page_Load事件处
理程序
    If Not Page.IsPostBack Then
        PrevText.Text="Prev"
        NextText.Text="Next"
        PageNumeric.Checked=True
        PageRows.Text="10"
        MyDataGrid.CurrentPageIndex=0
        BindDataGrid()
    End If
    End Sub

    Sub BindDataGrid() '向DataGrid控件绑
定数据源的事件处理程序
        MyDataGrid.PageSize=CInt(PageRows.
Text)
        If PageNumeric.Checked=True Then
'数字按钮样式的分页模式
            MyDataGrid.PagerStyle.
Mode=PagerMode.NumericPages
        Else
            '文本按钮样式的分页模
式
            MyDataGrid.PagerStyle.
Mode=PagerMode.NextPrev
        End If
        MyDataGrid.PagerStyle.
NextPageText=NextText.Text
    
```

```

MyDataGrid.PagerStyle.
PrevPageText=PrevText.Text
        MyDataGrid.
DataSource=CreateDataView()
        MyDataGrid.DataBind()
    End Sub

    Sub Page_Changed(Src As Object, E As
DataGridPageChangedEventArgs)
        '当用户单击某页分页连接按钮时执
行,处理PageIndexChanged事件
        MyDataGrid.CurrentPageIndex=E.
NewPageIndex
        BindDataGrid() '重新创建DataSet对象
并将其数据源绑定到DataGrid控件
    End Sub
</script></head>
<body><h3><center>轻松实现WEB数
据库的分页显示示例</center></h3>
<form runat="Server">
    连接到上一页时使用的文本: <ASP:
TextBox id="PrevText" runat="Server"/>&nbsp;
    连接到下一页时使用的文本: <ASP:
TextBox id="NextText" runat="Server"/><br>
    分页显示的样式:
    <ASP: RadioButton id="PageNumeric"
GroupName="Style" runat="Server" />数字
&nbsp; &nbsp;
    <ASP: RadioButton id="PageText"
GroupName="Style" runat="Server" />文本
<br>
    每页显示记录集的行数: <ASP:
TextBox id="PageRows" runat="Server" /> <p>
    <ASP: DataGrid id="MyDataGrid"
runat="Server" '声明一个DataGrid控件
        Width="90%"
        AllowPaging="True"
        PagerStyle_Position="Bottom"
        PagerStyle_HorizontalAlign="Right"
        GridLines="Vertical"
    
```

```

ONPageIndexChanged="Page_Changed"
/>
</form></body></html>

```

4 结束语

事实上, DataGrid控件控制页面导航标志显示风格的属性还有其他一些, 这里没有用到。用户可以查阅相关资料, 进一步设置页面的显示属性以满足实际需要。现在网站基本上都提供了分页显示信息的功能, 但大部分还是基于ASP的; 另一方面, 在以往的Web技术中, 经常是要编写一长串代码才能实现分页, 而且每用一个页面, 就要重写一次。在.NET框架的强大冲击下, 可以断言, ASP.NET必将成为新一代网站开发的主流技术, 因此本文讨论的技术很有实用价值。

参考文献

- 1 Jason Bell Mike Clark 等著, 赵颜敬译, ASP.NET 程序员参考手册, 清华大学出版社, 2002年5月。
- 2 Richard Anderson Brian Francis 等著, 王毅、杨浩等译, ASP.NET高级编程, 清华大学出版社, 2002年1月。
- 3 陈湘编著, ASP.NET与网站开发编程实战, 清华大学出版社, 2002年2月。