

# 开发基于 Web 的数据库应用程序

## Building Database Application Program Basing on Web

张仁平 周庆忠 (重庆后勤工程学院 400016)

**摘要:**本文在介绍开发基于 Web 的数据库应用程序常用对象或组件的基础上,介绍了开发基于 ISAPI 的 Web 应用程序的环境配置,最后程序实现了两个基于 Web 的数据库应用程序。

**关键词:**Web 应用程序 数据库应用程序 ISAPI NSAPI

### 1 引言

当前在 Windows 视窗平台中开发 Web 应用程序,比较流行的技术有 ASP.NET、VBScript/JavaScript、HTML 等,这些技术是 Microsoft 公司针对 Web 应用程序的开发而提出的,尤其是 ASP.NET 技术得到广泛应用。当采用这些技术开发基于 Web 的数据库应用程序,都需要在 Web 服务器建立 ASP 对象、ADO 对象来存取数据,这些要求程序员同时使用多种不同技术,而 Delphi 可以独立创建基于 Web 的数据库应用程序,只需要程序员会使用 Delphi 即可。这得力于 Delphi5 以上版本都具有强大的 InternetExpress 功能,它采用 WebBroker 技术,封装了大多数常用的 Web API 函数,加入扩展标识语言(Extensible Markup Language 简称 XML)标准封装数据。用 Delphi5 创建的基于 Web 的数据库应用程序可以在 IIS、PWS 或 Netscape 等 Web 服务器上正确运行。

### 2 常用对象或组件

开发基于 Web 的数据库应用程序常用以下对象或组件

#### 2.1 TWebModule 和 TWebDispatcher

TWebModule 对象是一个容器,它从 TDataModule 继承下来,包含响应客户的 HTTP 请求的所有逻辑。它只能放入非可视组件,Delphi 基于 TWebModule 创建的项目,是一个 DLL,其主单元声明了一个 TWebModule 对象。TWebModule 对象有个 Actions 属性,它是包含所有 TWebActionItem 对象的集合,点击 Actions 属性的省略号就会打开 WebModule Actions 对话框单击“New”按钮,就会新建一个 TWebActionItem 对象,然后可以在 OnAction 事件中添加相关源代码,完成某项特定功能。

如果要利用已有的 TDataModule 封装业务规则,就需要添加 TWebDispatcher 组件,用于接受 Web 服务器的请求,激活相关的 API 函数以动态的方式建立 HTML。

#### 2.2 TXMLBroker 和 TMIDASPageProducer

开发分布式 Web 应用程序离不开 TXMLBroker 组件和 TMIDASPageProducer 组件,TXMLBroker 组件主要用于从数据集中获得数据,并且以 XML 开工封装数据,TMIDASPageProducer 组件的主要作用在于根据 TXMLBroker 组件传递过来的数据,结合自身属性设置的用户界面,在浏览器中显示 XML/HTML 内容,并且向 Web 服务器返回用户请求、数据更新等信息。

TXMLBroker 组件的常用属性有三个:一是 Connect 属性,用于设置是否与 Web 服务器保持连接,通常设置为 true;二是 ProviderName 属性,用于设置提供数据集的对象 TProviderName,(TProviderName 组件发挥相当于 TDataSource 组件的功能);三是 ReconcileProducer 属性,用于选择设置 TMIDASPageProducer 组件。主要事件有 After/beforeDispatch:在 TXMLBroker 组件接受用户 HTTP 请求之前/之后触发,OnGetErrorResponse:在 TXMLBroker 更新数据发生错误时触发,OnGetResponse:在 TXMLBroker 更新数据成功时触发该事件,OnRequestRecord:在 TXMLBroker 向 TProviderName 组件获取数据时触发该事件,OnRequestUpdate:在 TXMLBroker 需要更新数据时触发。

TMIDASPageProducer 组件的常用属性有二个:一是 HTMLDoc 或 HTMLFile,用来指定 HTML 模板;二是 WebPageItems,用于打开组件编辑器,便于程序员在集成开发环境中设计网页。主要事件有 OnAftre/BeforeGetContent:在 TMIDASPageProducer 组件生成网页之后/之前触发,OnAfter/GetXMLData:在 TMIDASPageProducer 组件从 TXMLBroker 组件获取 XML 封装数据之后/之前触发该事件,OnHTMLTag:在 TMIDASPageProducer 组件所产生的网页中包含自定义的标记时,触发该事件来处理这些自定义标记。

#### 2.3 TPageProducer、TDataSetTableProducer 和 TQueryTableProducer

TPageProducer 组件、TDataSetTableProducer 组件和 TQueryTableProducer 组件都是从 TCustomContentProducer 继承下来,而 TCustomContentProducer 是一个抽象类,其作用是处理和操作 HTML 网页。如果需要生成简单的 HTML 网页,则只需 TPageProducer 就行了,如果要创建基于 Web 的数据库应用程序,那么就需要使用 TDataSetTableProducer 组件或 TQueryTableProducer 组件,与 TTable 组件或 TQuery 组件建立关联,获取数据建立 HTML 网页。

TPageProducer 组件的用法比较简单,它需要事先建立自定义标记,然后用实际内容替代这些标记,相当于给这些标记赋值。当 TPageProducer 组件的 Content 属性被设置“赋值给”TISAPIResponse 对象的 Content 属性时,就会在指定的 HTML 模板中查找标记,找到一个标记就触发一次 OnHTMLTag 事件。一般而言,HTML 模板中有多个标记,就要用 CASE 语句来进行处理。其 HTMLDoc 属性和 HTMLFile 属性用来指定 HTML 模板,既可以在编辑程序时指定 HTML 模板,也可以在程序运行时动态赋值;其 Dispatcher 属性用于指定 Web 应用程序发布接口,通常指定为 TWebModule 对象。

需要注意的是,虽然 Delphi 中定义一些默认的标记,并且这些标记也有特定含义,比如 tgTable 标记主要用 HTML 表格 <TABLE>..</TABLE> 来替换,但是,这些标记的真正用途取决于怎样使用它们,只要是自定义的标记,其值都是 tgCustom。

TDataSetTableProducer 组件和 TQueryTableProducer 组件的区别就象 TTable 组件和 TQuery 组件的区别,它们访问数据的方式不同,其共性在于它们的角色都非常类似于 TDBGrid 组件,通过其 DataSet 属性或 Query 属性指定一个 TDataSet 对象(TTable 组件或 TQuery 组件),这样就可以访问各种数据集建立 HTML 表格。

#### 2.4 TWebRequest 对象和 TWebResponse 对象

TWebRequest 对象和 TWebResponse 对象都是封装了 HTTP 协议的抽象类,它们都在 HTTPAPP.PAS 单元中声明,其中,TWebRequest 对象封装了用户向 Web 服务器发出的请求信息,包含客户相关信息,也可能包含许多空信息,因为客户的每个请求不一定完整;而 TWebResponse 对象则封装了 HTTP 协议允许的属性和方法来响应和处理客户的请求。

### 3 开发基于 ISAPI 的 Web 应用前的准备工作

在开发基于 ISAPI 的 Web 应用前需要作一些准备工作。首先是安装配置 Web 服务器,也就是建立相应的软件开发环境,进行相应的配置,其次是设置数据库别名。

下面,我们以 IIS 服务器为例,当然 Windows98 的用户可以采用 PWS,IIS 可以在 Windows2000Server 光盘中找到。如果没有可以到 Microsoft 的官方网站云下载。其次是 IIS 的配置,由于其配置比较复杂,下面详细介绍其配置步骤。

(1) 打开管理工具窗口。点击“开始”“设置”“控制面板”菜单进入控制面板,双击“管理工具”,进入“管理工具”窗口。

(2) 打开 IIS 管理程序。双击“Internet 服务管理器”图标,启动 IIS 管理程序。

(3) 新建虚拟目录。双击左边一栏中的“Internet 信息服务”打开该接点,依次打开“\*本机名”“默认的 Web 站点”,鼠标右击“默认的 Web 站点”项,在弹出的菜单中选择“新建”“虚拟目录”,打开“虚拟目录创建向导”对话框。

(4) 输入虚拟目录名。单击“下一步”进入“虚拟目录别名”设置对话框,在“别名”框中输入虚拟目录名,如 PP。

(5) 设置虚拟目录路径设置。单击“下一步”进入“目录路径”设置对话框,在“目录”框中输入物理目录名(或者点击右边的“浏览...”按钮,在弹出的“浏览文件夹”对话框中选择)。该目录要设置为调试下面两个 Web 应用程序示例所在文件夹,即输入“C:\P2”。

(6) 设置访问权限。单击“下一步”进入“访问权限”设置对话框,全部选择,点击“下一步”,进入“完成”对话框,点击“完成”按钮,完成设置。

(7) 设置执行许可协议。鼠标右击刚刚建立的虚拟目录 PP,弹出虚拟目录 PP 属性设置对话框,点击执行许可协议的下拉菜单,选择脚本和可执行程序,如图 1 所示。这一步非常关键也是初学者最容易忽略的地方,设置完这一步,才真正完成开发 Web 应用程序的环境配置,当然还可以进行其他属性设置,比如,点击图 1 中的“配置”按钮,可以选择脚本语言、会话时间、应用程序调试等等。

配置完 IIS 服务器后,建立一个指向“C:\P2”的数据库别名 P1。

### 4 开发一个简单 Web 应用程序

开发 Web 应用程序主要用到 Microsoft 公司和 Netscape 公司提供的 Web 服务器,它们根据自己的 Web 服务器提出了接口 API 函数即 ISAPI 和 NSAPI。ISAPI 是 Microsoft 公司提供的 Web 应用程序接口,是 Internet Server Application Programming Interface 的缩写,适

用于其 IIS 或 PWS 服务器,而 NSAPI 则是 Netscape 公司推出的 Web 应用程序接口,是 Netscape Application Programming Interfacier 的缩写,适用于其 Netscape 服务器,较 ISAPI 要复杂一些。Delphi 可以分别通过 ISAPI、PAS 单元和 NSAPI、PAS 单元访问它们。

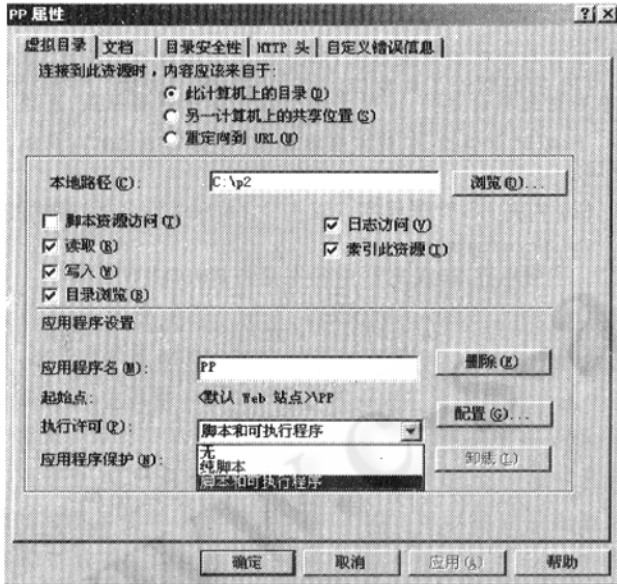


图 1 设置虚拟目录属性

开发具有数据访问功能的 Web 应用程序的主要步骤、方法如下:

(1) 创建一个包含 TWebModule 单元的新项目:点击“File\New”菜单,打开 New Items 对话框,双击 Web Server Application 图标,打开 Web 服务器扩展类型选择向导。三种类型分别为 ISAPI/NSAPI、CGI 和 Win CGI,通常选择 ISAPI/NSAPI。Web 应用程序类型选择完毕后,Delphi 就创建一个空白的包含 TWebModule 单元的新项目,该项目是一个 DLL。

(2) 配置 Web 数据模块:在建立新的项目即 Web 数据模块后,下面开始放入与数据相关的组件,建立 XML/HTML 内容了。首先放入一个 TSession 组件,并把它的 AutoSession 属性设置为 true。

注意:当我们需要创建基于 Web 的数据库应用程序时,必须把 TSession 组件放入 TWebModule 对象单元中。这主要是由于当多个用户同时向 Web 服务器发出请求时,Web 服务器需要为每个请求创建相应一个新的线程来运行 ISAPI 或 NSAPI 程序,因此,应用需要拥有自己唯一的 BDE 会话期,从而不与其他用户发出的请求发生冲突,这就要求把 TSession 组件的 AutoSessionName 属性设置为 true。

继续放入 TDataBase 组件、TQuery 组件或 TTable

组件、TDataSetProvider 组件、TXMLBroker 组件和 TMidasPageProducer 组件,将 TDataBase 组件的 AliasName 属性设置为 P1,即连接到 P1 数据库;设置 TQuery 组件的 SQL 属性置为“Select \* from 员工表”,并把 DataBaseName 属性设置与 TDataBase 组件的 DataBaseName 属性值一样,通过 TDataBase 组件连接到 P1 数据库;把 TDataSetProvider 组件的 DataSet 属性设置为 TQuery 组件的名字;把 TXMLBroker 组件的 Connect 属性设置为 true,把 ProviderName 属性设置为 TDataSetProvider 组件的名字,这样就可以访问员工表的内容,并对员工表的内容以 XML 形式进行封装;之后就是设置 TMidasPageProducer 组件建立网页显示内容。

双击 TMidasPageProducer 组件或点击“WebPageItems”属性,打开其组件编辑器,选中 TMidasPageProducer 组件,鼠标右击弹出对话框,选中“New Component”项,弹出添加 Web 组件对话框,选择组件类型。

由于我们需要浏览员工表的内容,所以首先选择 DataForm 组件,相当于数据窗体,浏览数据。接着向 DataForm 中加入显示数据的 GUI 组件即可视组件,选中 DataForm 鼠标右击,选中“New Component”项,会弹出“Add Web component”窗体,该窗体中列出所有可以加入到 DataForm 中的 Web 组件,分别选中 DataGrid 组件和 DataNavigator 组件。将 DataGrid 组件的 XMLBroker 属性设置为 TXMLBroker 组件的名字,将 DataNavigator 组件的 XMLComponent 属性值设置为 DataGrid1,这样 DataGrid 和 DataNavigator 就与 TXMLBroker 组件连接起来,可以浏览员工表的内容了。这样就基本完成 Web 应用程序了,剩下的工作就是把网页内容传送到 Web 服务器,当 Web 服务器得到这个 Web 应用程序时,就会以 HTML/XML 的形式发送到客户端的浏览器。要向 Web 服务器发送网页比较容易,选中 WebModule 组件的 Actions 属性,用鼠标右击它,点击“Add item”选项,生成一个新的 WebActionItem,设置其 Default 属性值为 true,设置其 Producer 属性值为 TMidasPageProducer 组件的名字。

(3) 编译、调试、运行 Web 应用程序:完成以上操作后就可以测试浏览这个 Web 应用程序了,需要注意的是,在编译之前,最好将编译完成的 Web 应用程序直接分发到前面新建的虚拟目录中。具体操作是点击“Project\Options”菜单,选中“Directories\Conditionals”选项卡,点击“Output Directory”右边按钮,选中虚拟目录“C:\P2”。点击“Project\Compile”菜单生成一个 DLL。这样就可以通过 URL 来浏览它了。

在编译、调试和运行 Web 应用程序前需要注意的有两

点:一是在编译、调试 Web 应用程序之前,不管使用什么 Web 服务器,当使用 Delphi 建立或调试 Web 应用程序时,甚至是要删除 Web 应用程序,必须要首先关掉 Web 服务器,然后才进行这些操作,否则系统会弹出操作错误信息。这是因为 Web 应用程序基本都是 DLL 文件或 EXE 文件,在 Web 服务器运行的时候就已经被调入内存,而 Windows 系统不允许重新编译或删除一个正在执行的文件。二是在 Web 应用程序运行之前,需要把“...Delphi5\Source\Web-Midas”目录下的所有文件拷贝到前面定义的虚拟目录 PP 中,即“C:\P2”目录中,否则无法浏览员工表的内容。

## 5 开发分布式 Web 应用程序

开发分布式 Web 应用程序与开发其他分布式应用程序相似,都需要先开发应用程序服务器,然后通过应用程序去访问服务器。下面先开发服务器应用程序。

### 5.1 开发服务器应用程序

点击“File\New”菜单,打开“New Items”窗体,选中“Multitier”选项卡,双击“Remote Data Module”图标,弹出远程数据模块设置向导,在 CoClass Name 栏输入远程数据模块的名字 MyServer。然后放入 TSession 组件、TDataBase 组件、TQuery 组件、TDataSetProvider 组件。分别设置 TSession 组件的 AutoSessionName 属性设置为 true,设置 TDataBase 组件的 AliasName 属性设置为 P1,DataBaseName 属性设置为 PPP,Connected 属性设置为 True,设置 TQuery 组件的 SQL 属性置为“Select \* from 员工表”,设置 DataBaseName 属性设置与 TDataBase 组件的 DataBaseName 属性值一样为 PPP,设置 Active 属性为 true,通过 TDataBase 组件连接到 P1 数据库,TDataSetProvider 组件的 DataSet 属性设置为 Query1,打开与员工表连接的通道。

在服务器应用程序主窗体中放入一个标签,其 Caption 属性为“这是我的服务器应用程序”,窗体的 Caption 属性为“Mysever Program”,以示区别。这样就完成服务器应用程序了,编译运行服务器应用程序后,下面开发客户端的 Web 应用程序。

### 5.2 开发客户端的 Web 应用程序

与前面一样,点击“File\New”菜单,打开 New Items 对话框,双击 Web Server Application 图标,打开 Web 服务器扩展类型选择向导。既可以选择 CGI Stand-alone executable,也可以选择 ISAPI NSAPI,在前面例子中我们已经实现了后者,这次选择前者选项,即建立 CGI 类型的 Web 应用程序。选择完毕后 Delphi 会创建一个空白的 Web 数据模块,我们的任务就是把 Web 数据模块与前面的服务器

应用程序建立通信。

放入一个 TDCOMConnection 组件,设置其 ComputerName 属性为 Zhang(我的计算机名也是执行服务器应用程序的计算机名),设置 ServerName 属性为 Sever-Project.MyServer,即前面的服务器名称后,将 Connected 属性设置为 true,即建立了应用程序与服务器之间的通信。

继续放入 TXMLBroker 组件和 TMidasPageProducer 组件,首先设置 TXMLBroker 组件的 RemoteServer 属性值为 DCOMConnection1,接着设置其 ProviderName 属性为 DataSetProvider1,即服务器中 TDataSetProvider 组件名,设置 Connect 属性为 true,TXMLBroker 就会激活服务器应用程序,并从服务器应用程序中取得数据,即员工表的内容了,然后把员工表的内容以 XML 格式封装起来,等待进一步处理;双击 TMidasPageProducer 组件或点击“WebPageItems”属性,打开其组件编辑器,加入一个 DataForm 窗体、一个 TDataNavigator 组件、一个 TDataGrid 组件,如同前面例子中的操作一样。

若用户要对数据表的字段名用中文显示,则可象一般数据库应用程序一样,鼠标右击 TDataGrid 组件,选中“Add All Fields”菜单,加入所有数据表字段名。然后选中每个数据字段名,将其 Caption 属性设置为相应的汉字名称。除此之外,还可以设置 TDataGrid 组件的 DisplayRows 属性,确定每页显示的数据记录行数,默认值是 4;还可以设置 HeadingAttributes 属性、RowsAttributes 属性和 Table Attributes 属性,对表头、表体和表框进行修饰处理。

最后需要把我们设计好的 Web 服务器应用程序回传给客户端浏览器。象前面操作一样,需要给 Web 服务器应用程序定义一个 WebActionItem。用鼠标右击 Web 数据模块的 Actions 对象,点击“Add item”选项,生成一个新的 WebActionItem,设置其 Default 属性值为 true,设置其 Producer 属性值为 TMidasPageProducer 组件的名字。这样就可以回传我们前面生成的图形用户界面内容了。

将 Web 服务器应用程序编译并分发到虚拟目录 PP 中,让客户端浏览器到存取它。当运行浏览器并且使用 UR 存取这个 Web 服务器应用程序时,我们就可以看到做好的 Web 应用程序了。

## 参考文献

- 1 Steve Teixeira & Xavier Pacheco, Delphi5 开发人员指南,机械工业出版社,2001 年 10 月。
- 2 李维,Delphi5.X 分布式多层应用电子商务篇,机械工业出版社,2000 年 5 月。