Delphi 数据库访问部件的事件处理

李秀川　（中央民族大学计算机系　100081）

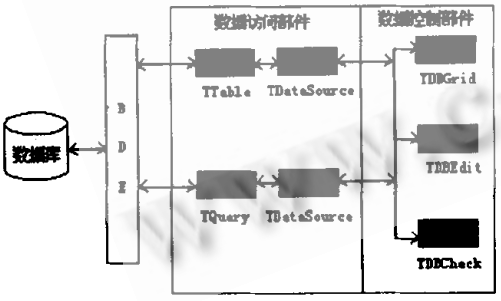
摘要：本文针对数据库开发中的实际情况，运用 Delphi 数据库访问部件的事件解决了在数据显示、处理中的一些技术问题。并以两个部件的三个重要事件为例作了分析。

关键词：BDE　部件　事件　属性

1. Delphi 数据库的体系结构

Delphi 是基于 Windows 环境的可视化的面向对象开发环境，也是当前最常用的 C/S(客户机/服务器)数据库开发工具。在数据库支持上，Delphi 以 BDE（Borland 数据库引擎)及 ODBC 为基础支持各种本地和远程的数据库系统。它采用可视化的构件实现对数据库的支持，为利用高级语言编写数据库应用程序提供了方便。

Delphi 利用数据访问部件（如 TDataSource、TTable 和 TQuery 等)直接访问数据库中的数据表以及访问表中的某些字段；而利用数据控制部件（如 TDBGrid、TDBEdit 和 TDBCheck 等)显示和浏览数据库的信息，为用户提供可视的界面对信息进行有效的浏览、编辑、插入和删除等操作。一般情况下，TdataSource 部件用来连接数据访问部件与数据控制部件。Delphi 数据库部件的体系结构如图所示。



2. 处理 TDBGrid 部件的 KeyPress 事件

利用 TDBGrid 部件以列表的形式来显示和编辑数据表是最常用的操作数据库的方式。但当在 TDBGrid 部件中进行编辑时，从一列转到另一列时需要按下 TAB 键才行。而在输入过程中，利用回车键更方便，也更符合使用习惯。如何才能把回车键转化为 TAB 键呢？对 TDBGrid 部件的 KeyPress 事件进行处理可达到这一目的。在下例中，假设在 Forml 中加入了一个类型为 TDBGrid 的 DBGridInput，DBGridInput 通过一个 DataSource 部件与数据表相连。源代码如下：

```
//处理 OnKeyPress 事件
procedure TForml.DBGridInputKeyPress(Sender:TObject;var Key:Char);
begin
case Key of
    #13:
  begin
     PostMessage(Sender.Handle,WM_KEYDOWN,VK_TAB,0);
    end;
    end;
end;
```

这里通过 Windows 中的 PostMessage 函数来把回车（ASCII 码为 13)输入转化为一个 TAB 键的 KeyDown 消息，则 Windows 系统最后“接收”的是 TAB 键而不是回车键。

3. 处理 TField 部件的 GetText 和 SetText 事件

数据库应用当中，TTable 和 TQuery 等部件都包含一个不可见的 TField 类型的对象 Fields，它是一个串列表，对应于数据库表或一个查询结果的列或字段。Fields 是随着 TTable 和 TQuery 等部件活动状态动态地建立的，当数据库表关闭时，它也随之消失。可以利用 Fields Editor 建立数据库中某一个字段的 TField 部件，对这个部件的 GetText 和 SetText 事件进行处理就可以让字段按自己的要求显示。

笔者在开发一个财务软件时发现：当在 TDBGrid 或

其他控制部件中显示金额数据时按会计规则必须包含千分号和小数点，而编辑金额时，如果包含千分号则会出现因输入值不是合法的 float 形式而出现异常。要解决这一问题，可以通过处理 TField 的 SetText 事件完成。在实际应用中，金额数据有时要求以元的形式显示，有时又可能要求以万元的形式显示。具体按什么方式显示通过用户的设置决定，如可按一个逻辑变量(MoneyUnit)的值确定。解决这一问题可通过处理 TField 的 GetText 事件完成。

下面是笔者以一数据库表 PZ.DB 为例(它包括一金额字段 JE)来介绍其解决方法。

(1)在 Delphi 的 IDE 环境中先建立一个名为 Forml 的 Form 对象；

(2)建立一个名为 Table 的 Table 对象，并设置其 TableName 属性指向 PZ.DB 数据表；

(3)建立一个名为 DataSourcel 的 DataSource 对象，并把其 DataSet 属性设为 Table；

(4)建立一个名为 DBGridl 的 DBGrid 对象，并把其 DataSorrce 属性设为 DataSourcel；

(5)在 Table 上单击鼠标右键，在快捷菜单中选择“Field Editor”命令，并在弹出的字段编辑窗口中再次按下鼠标右键，然后在快捷菜单上选择“Add All Fields”命令；

(6)在字段编辑窗口中选中 JE 字段，并设置其属性 DisplayFormat 为,,,.00(按会计规则显示金额数据)；

(7)对 JE 字段的 GetText 和 SetText 事件进行如下处理：

```
//处理 OnGetText 事件
procedure TForml.TablelJEGetText(Sender:TField;
var Text:String;DisplayText:Boolean);
var
  TempStr:String;
begin
  if MoneyUnit then    begin
    TempStr:= FormatFloat('#,##0.00',Sender.
Value/10000);
    Text:= TempStr;
  end
  else
    begin
    TempStr:= FormatFloat('#,##0.00',Sender.
Value);
    Text:= TempStr;
  end;
end;
//处理 OnSetText 事件
procedure Tforml.TableJESetText(Sender; const Text:
String);
begin
  if (Text='') then
    Sender.AsString:='0.00'
  else
    Sender.AsString:='SubStr(Text);
end;
//SubStr 函数的作用是去除字符串中的千分号,
Function SubStr(TrimString: String): String;
var
  S : String;
  Position : Byte;
begin
  S:= TrimString;
  Position := Pos)',',S);    While Position>0 do
  begin
    Delete(S,Position,1);
    Position := Pos(',',S);
  end;
  Result := S;
end;
```

SetText 和 GetText 是一个相反的过程：要想按自己的意愿显示数据，可在 OnGetText 事件中进行规范处理。例如，一般情况下逻辑型字段值在显示时真为 True，假为 False。而实际中经常要求显示时生动一些，如真显示为√，而假为×。与上面的程序相似，对逻辑字段的 OnGetText 进行必要的处理即可轻易实现；而要对显示的数据信息进行正确处理，应利用 OnSetText 事件把显示数据改回原来(系统接受)的形式，以免出现错误。

(来稿时间：1999 年 1 月)