

# 计算机高级编程技术



## 第七章 数据控制组件





## 第7章 数据控制组件

- 开发数据库应用程序是为了与数据库信息进行交互。数据控制组件通过使用数据库引擎BDE实现与数据库数据信息之间的联系。利用数据控制组件与数据库信息之间进行交互。
- 因为数据控制组件的功能是浏览和编辑数据库中的数据，所以数据控制组件又被称为数据浏览组件。





# 第7章 数据控制组件

- 7.1 数据控制组件简介
- 7.2 TDBGrid组件介绍
- 7.3 TDBNavigator组件介绍
- 7.4 TDBText组件介绍
- 7.5 TDBEdit组件介绍
- 7.6 TDBMemo组件介绍
- 7.7 TDBImage组件介绍
- 7.8 TDBList组件介绍





# 第7章 数据控制组件

- 7.9 TDBComboBox组件介绍
- 7.10 TDBCheckBox组件介绍
- 7.11 TDBRadioGroup组件介绍
- 7.12 TDBLookupListBox组件
- 7.13 TDBLookupComboBox
- 7.14 TDBCtrlGrid组件介绍
- 7.15 TDBRichEdit组件介绍
- 7.16 TDBChart组件介绍





## 7.1 数据控制组件简介

- 数据控制组件有TDBGrid、TDBNavigator、TDBText、TDBImage等。应用程序通过这些组件可以浏览数据集中的数据信息，还可以编辑修改数据信息并将它们提交给数据库。
- 它们有时候也被称为数据感知组件（Data-Aware），因为这些组件如果与一个处于打开状态的数据集相连，它们会马上显示该数据集中的数据。





## 7.2 TDBGrid组件介绍

---

- ❑ TDBGrid组件的作用是一个将数据集记录显示在网格中，并且可以对其中的数据进行编辑修改。
- ❑ 在窗体中放置一个TDBGrid对象来显示和编辑一个数据集中的记录，应用程序可以插入、删除或编辑数据集中的数据，或者只是显示数据集中的数据记录，如图所示。





## 7.2 TDBGrid组件介绍





## 7.2 TDBGrid组件介绍

- 在运行时，用户可以在数据网格中移动，也可以插入、删除和编辑数据。

编辑的结果在移动记录指针到另一条记录或者结束应用程序时提交给数据库。

- 7.2.1 TDBGrid组件的重要属性
- 7.2.2 TDBGrid组件的重要事件
- 7.2.3 TDBGrid组件的应用举例







## 7.2.1 TDBGrid组件的重要属性

---

- 1. Columns
  - Columns属性是用来读取和设置网格中列的特性的。
- 2. DataSource
  - DataSource属性指明了TDBGrid组件中显示数据的来源。TDBGrid组件与TdataSource组件发生联系，从而引入与此TdataSource组件相连的数据集，显示其中的数据信息。





## 7.2.1 TDBGrid组件的重要属性

### □ 3. DefaultDrawing

- 将DefaultDrawing属性设置为True允许TDBGrid组件自动地在每一单元格中显示数据；而将DefaultDrawing属性设置为False会取消自动显示。

### □ 4. DragMode

- 使用DragMode属性来控制是否允许用户对数据单元进行拖放操作。





## 7.2.1 TDBGrid组件的重要属性

- 5. EditorMode
  - EditorMode属性用来控制是否允许编辑当前单元内容。
- 6. Fields和FieldCount
  - 通过Fields属性可以直接地访问作为网格中一列的数据集中的字段对象。列的序号从0开始，第二列使用序号1，依此类推。
  - FieldCount属性是用来记录网格中列的总数。所以Fields序号的范围是从0~FieldCount-1。





## 7.2.1 TDBGrid组件的重要属性

- 7. Options
  - Options中包含了一系列规定TDBGrid显示和编辑操作属性的选项。
- 8. ReadOnly
  - ReadOnly属性是用来标志网格是只用来显示记录，还是允许用户通过它来编辑数据的。
- 9. SelectedField
  - SelectedField属性说明了当前选择单元格的字段。





## 7.2.2 TDBGrid组件的重要事件

事件	含义
OnCellClick	在单击网格中的单元格时触发
OnColEnter	当焦点移动到网格中的另一个单元格时触发
OnColExit	当一个单元格即将失去焦点时触发
OnColumnMoved	当网格中的一列发生移动时触发
OnEditButtonClick	在单击网格中一个字段的编辑按钮时触发
OnTitleClick	在单击网格中一个字段的标题时触发





## 7.2.3 TDBGrid组件的应用举例

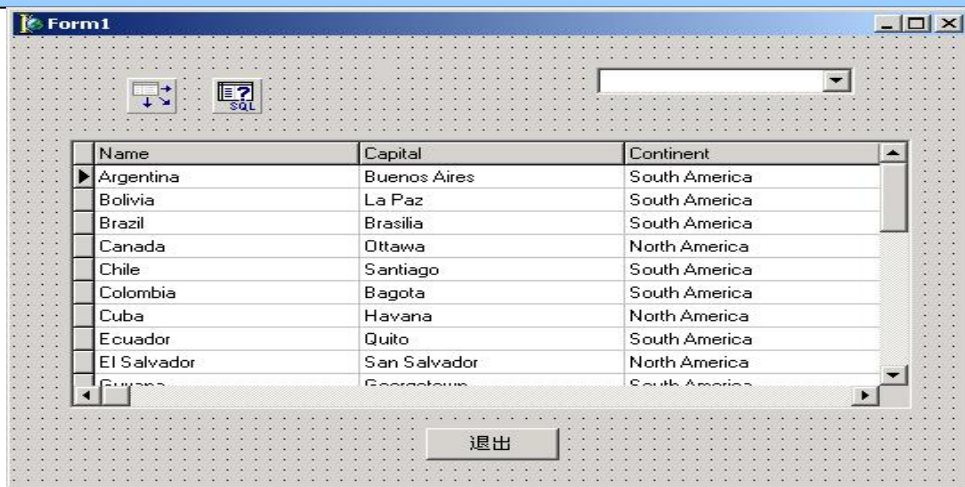
- **【例7-1】** 下面结合一个实际的例子来说明DBGrid组件的使用。
- 1. 窗体设计
  - 向窗体中添加组件，设置组件的属性后的窗体如图所示。
- 2. 编写程序代码
  - 该程序主要实现下面的功能：
    - 用户通过网格浏览数据库表格，可以对数据进行编辑修改。
    - 当用户双击单元格，显示其当前字段，并产生一个组合框。





## 7.2.3 TDBGrid组件的应用举例

- 用户在组合框中选取字段值，或自己输入字段值，单击后，自动添加到当前记录的对应字段中。





## 7.2.3 TDBGrid组件的应用举例

```
procedure TForm1.DBGrid1DbClick(Sender:TObject);
begin
  with DBGrid1.SelectedField do
    label1.visible:=true;
    label1.Caption:='请输入'+DisplayName+'值:.';
    with combobox1 do
      begin
        Visible:=true;
        items.Clear;
        case DBGrid1.SelectedField.Index of
          0: begin
              items.add('China');
              items.add('Japan');
              items.add('France');
            end;
          1:begin
              items.add('Beijing');
              items.add('Tokyo');
              items.add('Paris');
            end;
          2:begin
              items.add('Asia');
              items.add('Africa');
              items.add('Europe'); end;
```







## 7.2.3 TDBGrid组件的应用举例

```
3:begin
    items.add('100000');
    items.add('10000');
    items.add('1000');
end;
4:begin
    items.add('12000000');
    items.add('5000000');
    items.add('100000');
end;
end;
end;
end;
```

```
procedure
TForm1.ComboBox1Click(Sender:TObject);
var
    i:integer;
begin
    with combobox1 do
        begin
            query1.Edit;

            query1.FieldName(DBGrid1.SelectedField.
displayname).asString:=Items.Strings[ite
mindex];
```





## 7.2.3 TDBGrid组件的应用举例

---

```
query1.Post;  
  visible:=false;  
  label1.Visible:=false;  
end;  
end;
```





## 7.3 TDBNavigator组件介绍

- ❑ TDBNavigator组件（又称为数据库导航器）是用来在数据集中浏览数据和编辑数据的，例如插入一条空记录或者提交记录等。
- ❑ TDBNavigator组件是由一组控制按钮组成，用户可以完成在数据集中移动记录指针，增添或删除一项记录，编辑修改数据记录以及向数据库提交对数据集记录的修改等工作。
- ❑ 当用户使用TDBNavigator组件中的按钮时，与此TDBNavigator组件相连的数据集会发生相应的变化。





## 7.3.1 TDBNavigator组件实现的主要功能



名称	功能
First	记录指针指向第一条记录
Prior	记录指针指向上一条记录
Next	记录指针指向下一条记录
Last	记录指针指向最后一条记录
Insert	在当前记录前插入一条新记录
Delete	删除当前记录
Edit	编辑当前记录
Post	提交对当前记录的修改
Cancel	取消对当前记录的修改
Refresh	用数据集中的新记录刷新显示缓冲区





## 7.3.2 TDBNavigator组件的重要属性

### □ 1. ConfirmDelete

- ConfirmDelete属性用来决定用户在删除记录时是否显示要求用户确认的消息框。

### □ 2. DataSource

- 通过设置TDBNavigator组件的DataSource属性将它与其控制的数据集联系起来。





## 7.3.2 TDBNavigator组件的重要属性

- 3. ShowHint与Hints
  - ShowHint属性决定是否当鼠标在某一按钮上停留一段时间后，显示TDBNavigator组件中各按钮的提示信息。
  - Hints属性用来设置提示信息。
- 4. VisibleButtons
  - VisibleButtons属性用来确定TDBNavigator组件中需要显示哪些按钮。





## 7.4 TDBText组件介绍

- TDBText组件是用来显示数据集里当前记录中一个特定字段的值的数据控制组件。。
- TDBText组件显示的是当前记录中特定字段的值，所以随着记录指针的移动，其显示的内容也不断变化。
  - 7.4.1 TDBText组件的重要属性
  - 7.4.2 TDBText组件的应用举例





## 7.4.1 TDBText组件的重要属性

- 1. DataSource
  - DataSource属性，设置为一个数据源组件的名称，通过此数据源组件与数据集取得联系，从而读取数据集中的数据。
  
- 2. DataField
  - DataField属性是用来确定TDBText组件显示的是当前记录中哪个特定字段的值。







## 7.4.2 TDBText组件的应用举例

- **【例7-2】** 下面结合一个实际的例子来说明TDBText组件的使用。
- 1. 设计窗体
  - 窗体的设计如图所示。其中，将TDBText组件DataSource属性设置为窗体中的数据源组件名，DataField属性设置为数据集中的相应字段。





## 7.4.2 TDBText组件的应用举例





## 7.4.2 TDBText组件的应用举例

### □ 3. 编写应用程序的代码

```
procedure  
TForm1.Button1Click(Sender:T  
Object);  
begin  
  with Table1 do  
  begin  
    Open;  
    Setkey;  
  
    FieldByName('Name').AsString:  
    =Edit1.text;  
    If gotokey then  
    begin
```

```
label6.Caption:='查询成功!';  
dbtext1.Visible:=true;  
dbtext2.Visible:=true;  
dbtext3.Visible:=true;  
dbtext4.Visible:=true;  
end  
else  
begin  
  label6.Caption:='查询失败!';  
dbtext1.Visible:=false;  
dbtext2.Visible:=false;  
dbtext3.Visible:=false;  
dbtext4.Visible:=false;
```





## 7.4.2 TDBT<sub>ext</sub>组件的应用举例

```
end;  
end;  
end;  
procedure  
TForm1.Button2Click(Sender:T  
Object);  
begin  
    application.Terminate;  
end;  
procedure  
TForm1.DBNavigator1Click(Se  
nder:TObject;Button:TNavigate  
Btn);  
begin
```

```
label6.Caption:='正在浏览记  
录!';
```

```
table1.open;  
dbtext1.Visible:=true;  
dbtext2.Visible:=true;  
dbtext3.Visible:=true;  
dbtext4.Visible:=true;
```

```
edit1.Text:=table1.FieldByNam  
e('Name').AsString;  
end;
```





## 7.5 TDBEdit组件介绍

- ❑ TDBEdit组件是用来显示和编辑数据集里当前记录中一个特定字段的值的数据控制组件。用TDBText组件显示的字段值是只读属性的，而使用TDBEdit组件允许用户编辑修改数据。
- ❑ TDBText组件显示和编辑修改的是当前记录中特定字段的值。但与TDBGrid组件不同，它一次只能显示及编辑修改当前记录中特定的一个字段值。





## 7.5.1 TDBEdit组件的重要属性

### □ 1. DataSource

- TDBEdit组件的DataSource属性，设置为窗体中的一个数据源组件的名称，通过此数据源组件与数据集取得联系，从而对数据集中的数据进行操作。

### □ 2. DataField

- DataField属性是用来确定TDBEdit组件显示与编辑的是当前记录中哪个特定字段的值。





## 7.5.2 TDBEdit组件的应用举例

- 【例7-3】下面结合一个实际的例子来说明TDBEdit组件的使用。
- 1. 窗体设计

组件	属性名称	设置
Table1	DatabaseName	'DBDEMOS'
	TableName	'country.db'
DataSource1	DataSet	Table1
DBNavigator1	DataSource	DataSource1
DBEdit1	DataSource	DataSource1
	DataField	'Capital'
DBEdit2	DataSource	DataSource1
	DataField	'Continent'
DBEdit3	DataSource	DataSource1
	DataField	'Area'
DBEdit4	DataSource	DataSource1
	DataField	'Population'





## 7.5.2 TDBEdit组件的应用举例

A screenshot of a Windows-style application window titled "Form1". The window contains a toolbar with navigation and editing icons. Below the toolbar, there are several data fields with labels in Chinese: "国家名:" (Country Name) with the value "Bolivia", "首都:" (Capital) with "La Paz", "所在洲:" (Continent) with "South America", "面积:" (Area) with "1098575", and "人口:" (Population) with "7300000". A "查询" (Query) button is positioned to the left of the text "正在浏览记录!" (Currently viewing record!). At the bottom center, there is a "退出" (Exit) button.

国家名:	Bolivia
首都:	La Paz
所在洲:	South America
面积:	1098575
人口:	7300000







## 7.5.2 TDBEdit组件的应用举例

### 2. 编写程序代码

```
procedure
 TForm1.Button1Click(Sender:TObject);
begin
  with Table1 do
  begin
    Open;
    Setkey;

    FieldByName('Name').AsString:=Edit1.text;
    If gotokey then
```

```
label6.Caption:='查询成功!'
  else
  begin
    label6.Caption:='查询失败!';
    dbedit1.text:="";
    dbedit2.text:="";
    dbedit3.text:="";
    dbedit4.text:="";
  end;
end;
end;
```





## 7.5.2 TDBEdit组件的应用举例

```
procedure
 TForm1.Button2Click(Sender:
 TObject);
begin
  application.Terminate;
end;
procedure
 TForm1.DBNavigator1Click(S
  ender:TObject;Button:TNaviga
  teBtn);
begin
  with label6 do
    case Button of
```

```
nbFirst:Caption:='正在浏览记录!';
nbPrior:Caption:='正在浏览记录!';
nbNext:Caption:='正在浏览记录!';
nbLast:Caption:='正在浏览记录!';
nbInsert:Caption:='插入记录!';
nbDelete:Caption:='删除记录!';
nbEdit:Caption:='正在编辑记录!';
```





## 7.5.2 TDBEdit组件的应用举例

```
nbPost:Caption:='提交记录!';
  nbCancel:Caption:='取消
修改!';
  nbRefresh:Caption:='刷新
记录!';
end;
  table1.open;

Edit1.Text:=table1.FieldByNa
me('Name').AsString;
end;
procedure
TForm1.Edit1Click(Sender:T
Object);
begin
  label6.Caption:='正在查询记
录!';
end;
```





## 7.6 TDBMemo组件介绍

- TDBMemo组件是用来显示和编辑数据集中的多行文本的。TDBMemo组件允许用户显示和编辑一个包含多行文本内容的字段。TDBMemo的Text属性代表了该字段的内容。
- TDBMemo的显示与编辑数据集字段内容的功能与TDBEdit类似，只不过TDBMemo允许多行文本，并主要用于大二进制类型的文本对象（BLOBs），而TDBEdit组件显示较少内容的文本字段。





## 7.6.1 TDBMemo组件的重要属性

- TDBMemo组件的重要属性除了和其他数据控制组件一样的DataSource和DataField属性以外还具有以下不同的属性。
- 1. AutoDisplay
  - AutoDisplay是用来决定是否自动显示一个数据集中备注类型的字段值。
  - 当AutoDisplay属性值为True时，TDBMemo组件将随着当前记录的改变自动显示该字段的新的内容。
  - AutoDisplay属性只能应用于BLOB类型的字段。





## 7.6.1 TDBMemo组件的重要属性

### □ 2. Alignment

- Alignment属性是用来说明在TDBMemo组件中显示文本时的对齐方式。可以有左对齐，右对齐和居中三种方式。

### □ 3. MaxLength

- MaxLength属性用来限定可以输入TDBMemo组件中的最大字符数。此属性值为0时，表明不限制最大字符数。





## 7.6.1 TDBMemo组件的重要属性

### □ 4. WordWrap

- WordWrap属性是一个布尔值，用来指明向TDBMemo组件中输入的文本到了右边界处是否自动换行。

### □ 5. ReadOnly

- ReadOnly属性限定了TDBMemo组件中显示的文本是只读类型的，不能编辑修改。





## 7.7 TDBImage组件介绍

- ❑ TDBImage组件是用来显示和编辑当前记录中的BLOB类型的图形字段的。
- ❑ TDBImage组件还允许用户通过粘贴板对图形数据进行编辑操作。在转向其他记录前按Esc键可以返回该数据库图形的初始状态。
- ❑ TDBImage组件的重要属性与TDBMemo组件类似，它也是主要有DataSource属性、DataField属性和AutoDisplay属性。







## 7.8 TDBList组件介绍

- ❑ 在用户更改当前记录中特定字段值的时候，可以使用TDBListBox组件来提供给用户一个可选工程的列表。
- ❑ 列表框中的选项可以在设计阶段通过单击对象观察器的Items属性框激活String List editor进行编辑，也可以在程序运行时，使用Items的Add方法动态地改变。
- ❑ TDBList组件的重要属性有DataSource、DataField和Items属性。





## 7.9 TDBComboBox组件介绍

- ❑ TDBComboBox组件，是在用户更改当前记录中特定字段值时，用来提供给用户一个可选工程的下拉列表。
- ❑ TDBComboBox组件的重要属性有DataSource、DataField属性用来指向数据集中当前记录的特定字段。
- ❑ Items属性用来设置下拉列表的内容，和TDBList一样，组合框中的选项可以在设计阶段编辑，也可以在应用程序运行时动态改变。





## 7.10 TDBCheckBox组件介绍

- ❑ TDBCheckBox是允许用户选择或不选择一个值的数据控制组件，TDBCheckBox组件指向数据集中一个特定的字段，最适合于表达布尔类型数据的字段。
- ❑ 总而言之，TDBCheckBox组件表达一组包含两个可能值的数据。
- ❑ TDBCheckBox组件的DataSource属性和DataField属性将它与数据集的一个特定字段联系在了一起。TDBCheckBox组件主要是用来显示和修改一个布尔类型数据字段的字段值。





## 7.11 TDBRadioGroup组件介绍

- ❑ TDBRadioGroup组件是允许用户在一组选项中选取唯一一项的数据控制组件。TDBRadioGroup组件必须指向数据集中一个特定的字段。
- ❑ TDBRadioGroup组件由于DataSource属性和DataField属性将它与数据集的一个特定字段联系起来。主要是用来显示一个字段的字段值，以及编辑修改一个字段的字段值。





## 7.12 TDBLookupListBox组件介绍

- ❑ TDBLookupListBox组件是用来提供给用户一个数据集中的特定字段的字段值列表，它是自动显示数据集中指定字段的值。
- ❑ 实际上，TDBLookupListBox组件提供给用户的特定字段的字段值列表经常是来源于另一个相关的数据集。
- ❑ 因为TDBLookupListBox组件显示相关数据集的这种特性，它的重要属性还有KeyField、ListField、ListFieldIndex和ListSource这样的属性。





## 7.13 TDBLookupComboBox组件介绍

- ❑ TDBLookupComboBox组件是用来提供给用户一个数据集中的特定字段的字段值列表的，它的下拉列表的内容是另一个相关数据集中特定字段的字段值，它在设置好属性后自动添加到其中。
- ❑ TDBLookupComboBox组件的重要属性还有KeyField、ListField、ListFieldIndex和ListSource这样的属性。
- ❑ TDBLookupComboBox组件允许用户自行输入新的字段值用来编辑当前记录。





## 7.14 TDBCtrIGrid组件介绍

- TDBCtrIGrid组件其实和TDBGrid组件实现的功能差不多。只不过TDBCtrIGrid组件是用一种用户自己定义的方式来显示各条记录的。
- 在窗体中添加TDBCtrIGrid组件用来显示和编辑一个数据集中的记录。它允许用户使用自己定义的方式来显示各条记录。每条记录都显示在自己独立的Panel上。





## 7.14 TDBCtrGrid组件介绍

- ❑ TDBCtrGrid组件的属性与TDBGrid的属性类似，它也是显示数据集中记录的所有字段对象，所以没有DataField属性。但有一些可以让用户定义用于记录显示的Panel的属性。
- ❑ 在运行时，用户可以使用TDBNavigator组件在TDBCtrGrid中移动记录、添加或删除一条记录，还可以编辑修改当前记录，就像在TDBGrid中一样。







## 7.15 TDBRichEdit组件介绍

---

- TDBRichEdit组件的使用方法和它所实现的功能与TDBEdit组件非常相似，TDBRichEdit组件主要是用于RichEdit文本类型字段的显示和编辑。TDBRichEdit组件可以是多行显示。





## 7.16 TDBChart组件介绍

---

□ TDBChart组件由TChart组件派生而来，TDBChart组件与TChart组件的使用方法和实现功能类似，TDBChart组件有DataSource属性，用于数据集中数据汇总得到的图形。

