

利用 Delphi 制作不规则表格处理数据

王智军 (中国矿业大学能源利用与化学工程系 221008)

摘要:本文主要论述了利用 Delphi 制作不规则表格处理数据的机理及方法步骤。

关键词:DELPHI 控制点 同步移动 动态联接

Delphi 是目前最流行的开发 Windows 程序的软件之一,它具有基于窗口和面向对象的方法、高速的编译、强大的数据库支持和与 Windows 的紧密结合。它具有 Visual C++ 的功能、Visual Basic 的易于使用特性,因而倍受软件开发人员的青睐。

一、问题的提出

在开发与数据库相关的应用程序过程中,常需要用表格的形式来显示数据,并在表格中对数据加以处理。同时为了增强交互界面的友好性,最好能够实现类似于 Excel 电子表格的功能。在 DELPHI 中用数据库编程处理表格数据时应实现的基本功能包括:

1. 使表格中数据与数据库建立存取关系。
2. 可以在表格中对数据进行编辑。
3. 能将编辑过的数据存入数据库。
4. 能使表格随数据库中记录条数的变化而动态增删行或列。

国外使用的表格多为简单的一对一的形式,而在国内,各企事业单位则常制作一些形式复杂的表格来显示数据,以增加数据的对比性和透明度,便于进行分析和比较。其形式如图 1 所示。

图 1

在所有的编程语言中,制作图示表格都是一个较为

困难的问题。以前常用的方法是先用画图功能绘制一个表格,然后在表格的格栅中建立输入热域逐一添充数据。这种方法思路简单,易于实现,但用过的人都知道其存在不少缺陷,如每个变量与数据库建立存取关系相当复杂,同时形式固定单一,不能动态增删行或列。能否制作一个类似于 Excel 或 Word 中那样功能强大的表格来实现这些功能已摆在了大多数程序设计者面前。笔者经过一段时间的编程探索,摸索出了一个切实可行的方案。

二、DELPHI 处理复杂表格机理的分析

Delphi 提供了许多处理表格功能的工具。最常用的是 DBGrid 组件和 StringGrid 组件。

DBGrid 组件能直接与数据库建立动态联系,不需要编写代码就可直接实现用表格处理数据库中记录的功能,是一个理想的制作表格的工具。但使用单个该组件只能制作简单表格,无法实现上述复杂形式表格,每个该组件组合又不便于控制,特别是无法控制其中的滚动条。

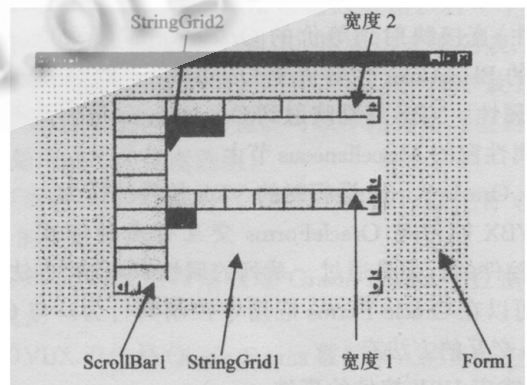


图 2

StringGrid 组件是纯粹的制作表格的工具。它不能直接与数据库建立联系,但该组件能显示任何形式的数据,不管数据来自何处,形式较为自由,可任意加以控制。

但是用单个的该组件仍无法实现上述复杂表格。

笔者在实践中摸索出了利用两个 StringGrid 组件和一个 ScrollBar 组件组合在一起制作复杂表格的方法。该表格能实现与数据库的动态联接,并可在表格中编辑数据。为便于阐明制作过程,现仍以图 1 中表格形式为例介绍如何实现其与数据库间的数据处理关系,同时在程序设计中给出了通用计算公式,适用于任意表格的处理。

要制作的表格形式如图 2 所示。

首先分析一下图 2 所示表格的特征。从整体上看可分为上、下两部分,都是规则的表格形式,同时上部分的一列对应下部分的两列,这样我们就可使用两个 StringGrid 组件,假设分别为 StringGrid1 和 StringGrid2,然后再添加一个 ScrollBar 组件来控制两个 StringGrid 组件的滚动。这时如何使两个 StringGrid 组件同步移动就成为问题的关键。处理不当将使两个表格不能同步移动,则会出现混乱,导致数据不能对齐。

要实现 StringGrid1 与 StringGrid2 的同步移动,必须有一个统一的控制点。可以用 StringGrid 组件的 LeftCol 属性作为控制点,对于一对二的表格形式,可采用如下公式来控制两个表格的移动:

$$\text{StringGrid1.LeftCol} := 2 * \text{StringGrid2.LeftCol} - 1$$

然而在实践中发现,用此方法实现起来较为复杂,且易出错。为此我们又选择了 ScrollBar 的滑块位置(Position)来控制两个表格的移动。具体原理是:

只要 ScrollBar 的滑块位置发生变化,则两个表格应同时移动。反之,若两个表格中的一个发生移动,则应设法使 ScrollBar 的滑块位置相应发生变化,从而带动另一个表格同步移动。

这样做可简化不少工作。解决了这个问题后,其他如鼠标、键盘事件的处理就容易多了。由于 Delphi 提供了功能强大的联接数据库控件 TTable 组件和 TDataSource 组件,使表格与数据库建立存取关系比较容易,在后面结合例子加以阐述。

三、用 DELPHI 实现复杂表格的具体方法与步骤

下面先介绍几个 TStringGrid 组件和 TScrollBar 组件的重要属性。

现以 StringGrid2 为例,简要介绍一下使用滚动条、TAB 键、箭头键及鼠标使两个表格同步移动的方法。

1. 布置窗体

拖动两个 StringGrid 组件和一个 ScrollBar 组件到 Form1 窗体上,各组件位置如图 2 所示。这里应当注意,

宽度 2 必须小于或等于宽度 1,否则在运行程序时,表格易发生混乱。

	属性	描述
TStringGrid	Col	表示当前具有输入焦点的行,最小值为 1
	Row	表示当前具有输入焦点的列,最小值为 1
	ColCount	用于设置列的个数,缺省值为 5
	RowCount	用于设置行的个数,缺省值为 5
	LeftCol	指定一个列显示在栅格的非固定列的最左边
	VisibleColCount	决定在栅格中完全可见的非固定列的个数
	Options.goTabs	用户可以使用 Tab 和 Shift+Tab 键移动
	Options.goEditing	用户可以直接编辑单元中的文字
TScrollBar	Position	设置或返回滚杆中小方块的位置
	Max	小方块滚动的最大值
	Min	小方块滚动的最小值,缺省值为 0

在设计阶段,布置好窗体后,应首先将 StringGrid1 和 StringGrid2 的 Options.goTabs 及 Options.goEditing 属性设为 TRUE。其他属性在 Form1 的 FormCreate 事件中设置,具体见所附程序清单。

2. 为 Form1 的 FormCreate 事件添加代码

为使小滑块滚动到最右边时,表中显示完全可视列,应使滚动条中小滑块位置的最大值设置如下:

$$\text{ScrollBar1.Max} := \text{StringGrid2.ColCount} - \text{StringGrid2.VisibleColCount};$$

为实现两个表格的对应关系,它们的列数关系表述如下:

$$\text{StringGrid1.ColCount} := (\text{StringGrid2.ColCount} - 1) * 2 + 1;$$

3. 滚动条事件处理

当用鼠标拖动小滑块或点击滚动条两侧箭头时,两个表格应同时滚动。相应事件为 ScrollBar1 的 OnChange 事件。为使小滑块的位置与表格的滚动建立对应关系,应使表格的 LeftCol 属性与小滑块的位置(Position)之间建立如下关系:

$$\text{StringGrid2.LeftCol} := \text{ScrollBar1.Position};$$

为使另一表格同步滚动,应使其 LeftCol 属性与 StringGrid2 的 LeftCol 属性建立如下关系:

```
StringGrid1.LeftCol := 2 * StringGrid2.LeftCol - 1;
```

由上述代码可以看出,当滚动条中滑块的位置(Position)发生变化时,两个表格也相应同向滚动。

4. 鼠标事件处理

当用鼠标点击表格内部时,输入焦点随之发生变化,若点击未完全显示列,则表格会发生滚动,这时应设法使滚动条的滑块也滚动到相应的位置,触发 ScrollBar1 的 OnChange 事件,从而带动另一表格同步滚动。这一动作发生在 StringGrid2 的 OnMouseUp 事件中。为实现这一点,表格的 LeftCol 属性与小滑块的位置(Position)之间应为如下关系:

```
ScrollBar1.Position := StringGrid2.LeftCol;
```

5. 键盘事件处理

当使用上下箭头键时,应能使输入焦点在两个表格对应列内上下移动;当用左右箭头键使输入焦点移动到显示区的最左边或最右边时,表格会滚动一列,应使滚动条的滑块同时也滚动到相应的位置,从而触发 ScrollBar1 的 OnChange 事件,带动另一表格同步滚动。使用 TAB 键的效果与使用右箭头键相同。所有这些动作都发生在 StringGrid2 的 OnKeyDown 事件中。

当用下头键(vk-down)使输入焦点向下移动,并且移动到最后一行时,应使输入焦点进入下一个表格。用下面的条件判断输入焦点是否已移动到表格的最后一行。

```
if StringGrid2.Row = StringGrid2.RowCount - 1 then
```

.....

下面的语句可使输入焦点进入下一个表格:

```
StringGrid1.SetFocus;
```

```
StringGrid1.Col := StringGrid2.Col * 2 - 1;
```

用上箭头键(vk-up)移动输入焦点的方法与用下箭头键相似,具体见所附的程序清单。

用右箭头键(vk-right)时,输入焦点向右移动,应使输入焦点的位置与滑块的位置相对应,它们的关系如下:

```
ScrollBar1.Position := StringGrid2.Col + 1;
```

用 TAB 键(vk-tab)的方法与右箭头键相同,只是当输入焦点移动到最后一列时,应使其返回到第一列。用下面的条件判断是否移动到最后一列:

```
if Stringgrid2.Col = StringGrid2.ColCount - 1 then.....
```

用左箭头键时(vk-left),输入焦点向左移动,同样应使输入焦点的位置与滑块的位置相对应,它们的关系如下:

```
ScrollBar1.Position := StringGrid2.LeftCol - 1
```

为 StringGrid1 添加事件代码的方法与 StringGrid2 相同,宗旨是当表格发生滚动时,应设法使滚动条的滑块位置(Position)相应发生变化,触发 ScrollBar1 的 OnChange 事件,从而带动 StringGrid2 的表格同步移动。

6. 与数据库连接的方法

要实现表格与数据库的动态联接功能则比较容易。假设已经建好一个数据库,将一个 TTable 组件和一个 TDataSource 组件加到 Form1 窗体上,设置好相关属性后,用循环语句将每条记录分别添加到表格中即可,保存数据则是它的逆操作。以 StringGrid1 为例,要在表格中显示数据,只需在 TForm1.FormCreate 事件中添加如下几行代码即可,代码如下:

```
with Table1 do
```

```
begin
```

```
Active := true; //打开数据库
```

```
First; //从第一条记录开始
```

for j := 0 to FieldCount - 1 do //将每一个字段名显示在表格内

```
StringGrid1.Cells[j, 0] := Fields[j].DisplayName;
```

for k := 1 to RecordCount do //将每一个字段值显示在表格内

```
begin
```

```
for j := 0 to FieldCount - 1 do
```

```
StringGrid1.Cells[j, k] := Fields[j].AsString;
```

```
next; //下一条记录
```

```
end;
```

```
Close; //关闭数据库
```

```
end;
```

要实现表格行与列的动态增删,以 StringGrid1 为例,只需在 TForm1.FormCreate 事件中添加如下代码即可。

```
StringGrid1.RowCount := Table1.RecordCount + 1;
```

```
StringGrid1.ColCount := Table1.FieldCount;
```

为 StringGrid2 添加代码的方法与 StringGrid1 相同。

用上述方法处理不规则表格与数据库的关系结构清晰、方法实用,完全适合于中国人的习惯。只要按照本文介绍的机理去做,表格的外观完全可由设计者在设计时随意调整,当然也可以在程序内部用代码加以控制。由于篇幅所限,未能尽述该表格的所有相关功能,如用 CellRect 方法合并栅格,就象 Excel 中合并单元格那样来处理局部栅格等。同时如何打印该表格也是一个值得探讨的问题。

完全理解了上述方法后,就可以自己制作更复杂的表格了。

(来稿时间:1998年10月)