

使用 Delphi 经验三则

杨世勇 (重庆大学自动化系 400044)

1. 用 ComboBox 构件实现数据控件 DBComboBox 的功能

由于 Delphi 的 DBComboBox 控件没有 Text 属性,使其在某些场合编程时很不方便。而普通 ComboBox 控件虽有 Text 属性却无 DataField 属性,笔者通过将 ComboBox 与 Query 控件的配合使用,使其成为既有 Text 属性又包含一个数据库表字段的下拉列表框。

实例:某数据库系统要求用下拉列表框对一个名单表格的“NAME”字段的内容进行选择,以便实现系统按“姓名”进行动态查询。见下表:

名单表 MD.DB

NUMBER	NAME
1	张 红
2	李 兵
...	...

这里我们用 ComboBox 配合 Query 控件来实现上述

功能的下拉列表框,具体实现方法很简单,首先用 Query 在表格 MD.DB 中查找到字段 NAME,然后将其内容加入到 ComboBox 控件中。

设窗体中已放置了 Query1 控件,在其 SQL 属性的 String list editor 窗口输入下列代码实现 NAME 字段查询:

```
select  
MD."NAME"  
from'e:/work/MD.DB' |e:/work 是存放数据库表格的路径|
```

然后在窗体的 OnShow 事件中输入下列代码,便可将字段 NAME 的内容加入到 ComboBox1 控件中:

```
procedure TForm1.FormShow(Sender: TObject);  
var  
XM: string; {定义一个暂存姓名的中间变量为字符串类型}  
begin  
XM := query1.fieldbyname('NAME').asString;  
combobox1.items.add(XM);
```

```

query1.next;
end;
query1.close;
end;

```

运行程序并显示 Form1 窗体后,若打开该下拉列表框,则其中的内容便是字段 NAME 的内容。

2. 用 Winexec() 函数来驱动其他程序

在许多应用程序中常常需要驱动其他的应用程序以增强本系统的功能。在 Delphi 中可通过调用 Windows API 函数 Winexec() 来驱动其他程序, Delphi 中 Winexec() 函数的格式为:

```

function Winexec ( CmdLine: PChar, CmdShow:
Word): Word;

```

其功能是执行指定的其他应用程序,它包括两个参数:CmdLine 和 CmdShow。CmdLine 指向一个以 NULL 结尾的 Windows 字符串,也即将要运行的应用程序命令行,包括可执行文件的路径和文件名以及可选参数(即在启动应用程序的同时需要打开的文件名);CmdShow 指定应用程序的窗口显示状态。

实例:某数据库系统的一个查询界面中,要求双击某 AutoCAD 图形文件名(显示在 DBEdit 构件内),系统便由 Delphi 切换至 AutoCAD,并恰显示该图形文件名对应的 AutoCAD 图形文件,在构件 DBEdit 的 OnDbClick 事件中输入下列代码,便能实现上述功能:

```

procedure TForm1.DBEdit1DbClick(Sender: TObject);
var
FileName: String; {文件名变量 FileName 定义为 Pascal 式
字符串类型}
WJName: array [0..30] of char {文件名变量 WJName 定义
为 NULL 式字符串类型}
begin
Filename:= 'd:/acadwin/acad' + table1.Fieldbyname('
TZWJM').AsString;
{d:/acadwin/acad 是 AutoCAD 可执行文件的路径及文
件名}
StrPCopy(WJName, FileName); {将 Pascal 式字符串转

```

化为 NULL 式字符串}

```

Winexec(WJName, sw-show); {调用函数且激活的窗口
以正常状态显示}
end;

```

说明:程序中 TZWJM(图纸文件名)是数据控件 table1 访问的数据库表格的一个字段,字段的内容是 AutoCAD 图形文件文件名(即切换至 AutoCAD 后所要打开图形的文件名)及其所存放的路径。

3. 用分页构件设计系统界面

Delphi 提供了三个分页构件,分别为 Tabset、Notebook 和 TabbedNotbook,前两个必需配合起来使用。合理地利用它们设计系统的界面,不但使界面美观、操作方便,有时还可以减少程序所占的储存空间。这里不打算举具体的实例,仅根据笔者的实际开发经验谈几点体会。

(1) 如果一个系统的界面很多,且某些界面的信息相似时,可以考虑将这些信息相似的界面进行同窗体的分布设计;

(2) 构件 Tabset 与 Notebook 的组合和构件 TabbedNotbook 在功能上相同,但使用后者更为简单,两者主要在界面的视觉效果上不同(页码名的位置不同),采用哪种构件应以使界面美观、操作方便而定;

(3) 构件 Tabset 和 Notebook 配合使用时,要注意以下两点:

- 设置 Tabset 的 Tabs 属性和 Notebook 的 Pages 属性时,输入顺序一定要对应,否则页码与页面就不对应;

- 为了将构件 Tabset 和 Notebook 连接,需在 Tabset 的 OnClick 事件中加入如下代码:

```

notebook1.pageindex:= tabset1.tabindex;

```

(4) 有些时候,希望窗体刚创建时就显示第一页界面,可以在窗体的 OnCreate 事件中加入下列代码进行初始化:

```

tabset1.tabindex:= 0;

```

```

notebook1.pageindex:= - 0;

```

(使用 TabbedNotbook 构件时的初始化代码为: tabbednotebook1.pageindex:= 0;)

(来稿时间:1998 年 6 月)