

在 Delphi 应用程序中提高录入效率的方法*

Tips for Raising Data Input Speed in the Delphi Program

杨 柳
Yang Liu

(广西大学计算机与信息工程学院 南宁 530004)
(College of Comp. and Info. Engi., Guangxi Univ., Nanning, 530004)

摘要 利用 Delphi 的 ComboBox 组件和系统初始信息、DBGrid 组件、代码和设置文本输入框的输入模式等4种方法,提高了录入效率。给出1个例程的详细代码。

关键词 Delphi 应用程序 录入效率

中图分类号 TP391.14 A

Abstract The efficiency of data input is raised by using ComboBox subassembly and initial information of system, DBGrid subassembly, codes, and setup of input mode of text box. A detailed codes of a sample is revealed.

Key words Delphi, application program, data input

1 问题的提出

在众多信息管理系统中,录入操作常常是不可避免的。有时需要重复录入某些数据,某些数据在录入时还有一定的限制,如必须是数字、字母,或只能限定在一定的取值范围之内。当数据的日常录入工作量很大时,录入操作的繁琐和低效,势必会给用户造成不便。

作者为南宁市某公司开发的物资管理系统以 Client/Server 二层体系结构模式作为系统开发平台,以 SQL Server 7.0 作后台数据库系统, Delphi 5.0 作为数据库系统前端开发工具。该系统满足了企业内部网络环境的分布式数据管理。但是,数据日常录入量较大,目前数据库中已录入各种不同的物资材料信息8174条,仅2001年需录入的材料出入库记帐数据就有几十万条。为了使用户在录入数据时能更高效、快捷,有必要设计一个简捷灵活的录入操作模式,尽量减少用户的击键次数,方便用户操作。

2002-06-08 收稿。

* 广西自然科学基金(桂科自9912009)资助项目。

2 Delphi 5.0的特点

Delphi 是 Inprise 公司推出的先进的可视化开发工具^[1],具有强大的面向对象开发的能力。Delphi 5.0的可视化开发环境,使开发人员只要在面板上选择相应组件就能快速建立应用程序和开发原型;用最新的 Frames 功能,开发者能够可视化地为商业逻辑创建组件。Delphi 5.0基于组件的技术大大简化了许多复杂的开发任务,成为新一代数据库开发工具的佼佼者。

3 在 Delphi 应用程序中实现高效快捷的录入操作的几种方法

结合 Delphi 5.0提供的功能,在开发物资管理系统时,作者设计如下4种录入操作方法来减轻用户的录入压力,提高用户的录入效率。

3.1 利用 Delphi 的 ComboBox 组件和系统的初始信息简化用户的录入操作

物资管理系统最常用的操作是物资材料出入库登记,在录入材料的明细账记录前,先要录入一些固定的初始信息,如存放材料的公司仓库名称、本次操作的人员姓名、供应厂商名称等等。Delphi 提供了一个下拉列表框组件:ComboBox,在下拉列表框中要显示的信息(如仓库名称)可以写在该组件的 Items 属性中;当用户进行材料的登记入帐等操作时,碰到需要录入诸如仓库名、操作员姓名或供应厂商名等信息的地方,就可以直接把这些信息显示在下拉列表框中,用户只需用鼠标选择相应名称就可完成该信息的录入,不必重新敲入一个长长的名称(特别是中文名称)。但是在 Delphi 窗体的设计期直接把仓库名称写在该组件的 Items 属性是行不通的,因为仓库名称并非固定不变,系统随时可以新增一个仓库,或对一个已有的仓库更名。因此,必须为 ComboBox 建立一个动态的下拉列表信息,该下拉列表信息与数据库中仓库表的仓库名称是一致的,当仓库名称改变时,该下拉列表框的信息也随之改变,实现仓库名称的动态显示,保证录入数据的一致性。实现方法如下:

(1)先在后台数据库中建立一个仓库名称表(表名设为 G_Name),专门存放仓库名称。

(2)在公共单元中创建一个过程,把名称表中的名称字段值读到数组 NameA 中,以后若用到该名称字段值时,可以直接从数组 NameA 导入,而不需每次都到后台调用数据库表,以减少网络的传输操作,提高效率。该过程定义如下:

```
procedure NameToArray(S1,S2:String);
var
  SQLCode:String;
begin
  SQLCode:= 'select '+s1+ 'from '+s2;
  ItemNum:=0; // ItemNum 为在公共单元定义的全局整型变量
  with DataModule1.ADOQuery1 do begin
    Close;
    SQL.Clear;
    SQL.Add(SQLCode);
    OPEN;
    SetLength(NameA,RecordCount); // NameA 为在公共单元定义的全局数组变量
    while not(eof) do begin
```

```

    NameA[ItemNum]:=FieldByName(s1).asString;
    Next;
    ItemNum:=ItemNum+1;
end;
Close;
end;
end。

```

(3)在公共单元再创建一个公用过程,将数组 NameA 的元素填入 ComboBox 组件的 Items 属性,该过程定义如下:

```

procedure ArrayToComboBoxItems(ComboBox1:TComboBox);
var i:integer;
begin
    ComboBox1.Items.Clear;
    for i:=0 to ItemNum-1 do
        ComboBox1.Items.Add(NameA[i]);
    End;

```

(4)在窗体 Form 的 OnActivate 事件中调用以上2个过程,代码如下:

```

procedure TForm1.FormActivate(Sender:TObject);
begin
    Unit1.NameToDwArray('库名','G-NAME');
    Unit1.ArrayToComboBoxItems(ComboBox1);
end。

```

这样一来,当需要在组件 ComboBox1 中录入仓库名称时,该组件的下拉列表框会显示出与后台数据库 G_Name 表一致的仓库名。

3.2 有效利用 Delphi 的 DBGrid 组件

利用下拉列表框极大地方便了用户的录入操作,但有些录入(特别是主从表数据的录入)要在 DBGrid 网格中进行。Delphi 提供功能强大的 DBGrid 组件,但是 DBGrid 不直接支持下拉列表框的输入^[2]。如可以利用 Delphi 的 DBLookupComboBox 组件,并书写 DBGrid 的 DrawDataCell 事件、OnColExit 事件和 KeyPress 事件的例程。其中 DBLookupComboBox 组件用来取得仓库名称表中的仓库名称,DrawDataCell 事件用来重新绘制 DBGrid 的单元格,OnColExit 事件规定 DBGrid 单元格在未获得焦点时不显示 DBLookupComboBox,然后在 KeyPress 事件中调用一个 Windows API 函数 SendMessage,将数据传输到 DBLookupComboBox 中,从而达到在 DBGrid 网格中利用下拉列表框进行数据录入的目的。

3.3 利用代码提高用户录入效率

提高用户录入速度的另一种方法是编码录入。这需要在数据库中设计一个代码表,如:建立了一个供应厂商代码表,将每个供应厂商的编号、名称及其对应的五笔代码和拼音代码存入代码表,用户录入时无论是录入编号、名称、还是五笔代码或拼音代码,都可以显示出该供应厂商的名称。如:当用户录入一个拼音码(或五笔码)时,系统显示出来的就是根据这个拼音码对应的供应厂商名。

在 Delphi 应用程序中将 SQL 语句:

```
Select 供应厂商名 from 代码表 where (编号=用户录入)  
or (拼音码=用户录入) or (五笔码=用户录入)
```

嵌入 Edit 组件的 OnExit 事件即可实现这个功能。

3.4 设置文本输入框的输入模式

用户在录入时,可能有些数据只能是数字(如数量、金额等),有些只能输入英文字母,有些则必须输入汉字。对必须输入汉字的文本编辑框(Edit 组件),将其 ImeMode 属性值设为“ImChinese”, ImeName 属性值设为“全拼输入法”或其他中文输入法。这些文本编辑框取得焦点后,系统会自动切换成用户设置的中文输入模式;失去焦点后,又会自动切换成英文输入模式,无需用户手动切换,非常灵活方便。同时,也可以设置一些输入的限制,如文本编辑框的 OnKeyPress 事件例程如下:

```
if not(key in ['0'..'9', #8]) then  
    key := #0;
```

可以限制用户在该文本编辑框中只能输入数字。如果想限制用户只能输入字母,则将文本编辑框的 OnKeyPress 事件例程写成:

```
if not(key in ['a'..'z', 'A'..'Z', #8]) then  
    key := #0;
```

这样设置的好处是,防止用户录入错误,可以减少录入操作时间,系统的容错性也得以提高。

4 结束语

高效、快捷、简便和性能的优化是每个信息管理系统追求的目标之一。在一个录入工作量大的信息管理系统中,录入操作的简便和快速是系统不得不考虑的问题。本文介绍的物资管理系统通过利用开发工具 Delphi 5.0 提供的功能特点和系统的初始信息,有效地解决了实际应用中的一些问题,提高了录入操作的效率。目前系统的代码已编制完成,正在试行阶段,用户对其中简便灵活的录入方式非常满意。

参考文献

- 1 谭 鹏. Delphi 5.0 数据库应用开发与编程技巧. 武汉:华中理工大学出版社,2000.
- 2 季雪岗,王晓辉,张宏林. Delphi 编程疑难详解. 北京:人民邮电出版社,2000.

(责任编辑:邓大玉)