

数据窗口的使用技巧

刘常杰 续 斌 (天津大学 300072)

1. 如何在数据窗口中用 Enter 键实现列跳转

在数据窗口中,列之间的跳转是用 Tab 键实现的。用户按下 Tab 键,数据窗口中列上的焦点会按数据窗口 Taborder 属性确定的跳转顺序跳到下一列上。但对于最终用户尤其是那些需要进行大量录入工作的用户,用 Tab 键实现列跳转无疑是很不方便的。为了解决这个问题,最常用的办法是定义一个数据窗口的用户事件,通过编程用 Enter 键实现列跳转。例如一个包含了四个字段:“编号”、“姓名”、“电话”、“通信地址”的数据窗口。我们在该数据窗口中定义一用户事件 ue-downkey,事件 ID 是 pbm-dwnkey,当数据窗口得到焦点,如按下任意键,该事件被触发。其脚本如下:

```
if keydown(KeyEnter!) then
  choose case dw-1.getcolumnname()
    case "编号"
      dw-1.setcolumn("姓名")
    case "姓名"
      dw-1.setcolumn("电话")
    case "电话"
      dw-1.setcolumn("通信地址")
  end choose
end if
```

此外,通过发 Windows 的 Tab 消息给 Windows,也是实现这一功能的有效途径。可在数据窗口中定义一用户事件 ue-downenterkey,事件 ID 是 pbm-dwnprocessenter,当数据窗口得到焦点,如按下 Enter 键,该事件被触发。其脚本如下:

```
Send(Handle(this), 274, 61472, 0)
```

需要注意的是,用 Enter 键实现数据窗口的列跳转会与窗口中的缺省命令按钮发生冲突。在这种情况下,建议将命令按钮的功能放在工具栏上,窗口中不要安排命令按钮。

2. 如何利用数据窗口的 DBError 事件

无论什么时候,由数据窗口的动作而引起的数据库错误发生时,都将触发 DBError 事件。在缺省情况下,此事件显示一个系统的错误消息窗口。该事件的事件参

数如下:

```
sqldbcode(long) DBMS 供应商指定的错误代码
sqlerrtext(string) DBMS 供应商指定的错误信息
sqlsyntax(string) 导致错误的 sql 语句
buffer(dwBuffer) 在错误发生时所使用的数据窗口
控件缓冲区
row(long) 数据窗口中引起错误的那一行
```

我们对该事件的处理方法通常是利用 Choose... Case 语句,根据系统返回的不同错误代码编写脚本,并在脚本的最后将该事件返回代码设置为 1,来达到显示定制的错误消息窗口并屏蔽掉系统的错误消息窗口的目的。

下面举例说明。

```
int li-startpos, li-endpos
string ls-errtext, ls-tag, ls-columnname
CHOOSE CASE sqldbcode
//错误号 - 195 表示数据库非空字段为空值
CASE - 195
//从系统返回的错误信息中取出发生错误的列名
li-startpos = pos(sqlerrtext, 'column')
li-endpos = pos(sqlerrtext, 'in')
ls-columnname = trim( mid( sqlerrtext, li-startpos +
8, &
li-endpos - li-startpos - 9))
//取出该列 tag 属性中的内容(tag 属性一般用于设置对
//象的提示信息或说明性文字。在此处其设置方法是这样
//的:打开数据窗口,双击该列,在弹出的 Column
Object
//对话框的 General 页面中可设置该属性)
ls-tag = ls-columnname + ".tag"
ls-errtext = describe(ls-tag) + "不可为空"
//弹出定制的错误信息窗口
messagebox("信息提示", ls-errtext)
//将焦点放在该列上
```

```
setcolumn(ls-columnname)
//对其他错误进行处理
...
END CHOOSE
//屏蔽系统错误信息窗口
RETURN 1
```

在这里也需要注意一点：对于相同的错误，不同 DBMS 下的 DBError 事件的 sqldbcode 和 sqlerrtext 也是不同的。上面例子的后台 DBMS 是 Sybase SQL Anywhere 5.0。如需将其移植到其他 DBMS 下，需要对程序稍做改动。上面例程在 MS SQL Server 6.0 下应改为：

```
int li-startpos, li-endpos
string ls-errtext, ls-tag, ls-columnname
CHOOSE CASE sqldbcode
CASE 233
```

```
li-startpos = pos(sqlerrtext, 'column')
li-endpos = pos(sqlerrtext, 'in')
ls-columnname = trim(mid(sqlerrtext, li-startpos + 7,
&
li-endpos - li-startpos - 7))
ls-tag = ls-columnname + ".tag"
ls-errtext = describe(ls-tag) + "不可为空"
messagebox("信息提示", ls-errtext)
setcolumn(ls-columnname)
//对其他错误进行处理
...
END CHOOSE
RETURN 1
```

(来稿时间:1998年9月)