

PB 中应用 Windows 联机帮助的设计与实现

董祥军 翟延富 张洁 (山东轻工业学院计算机系 250100)

摘要: 本文主要介绍了 Windows 联机帮助的设计与实现技术以及在 PowerBuilder 应用系统中使用 Windows 联机帮助的几种方法。

关键词: PowerBuilder 联机帮助 设计与实现

一个好的应用系统离不开帮助文件的支持,但如何编写有自己特色的 Windows 帮助文件,是标志一个系统是否好用的重要因素。下面介绍一下自己的体会。帮助文件是一个扩展名为 HLP 的标准文件,当我们打开(如双击它)一个这样的文件时,该文件会自动装载 WINHELP.EXE(一个旨在显示 HLP 文件的 Windows 应用程序),出现你熟悉的帮助界面。一般的开发环境都提供了应用 HLP 文件的方法,只要在自己的应用系统中挂接上自己的 HLP 文件,你就拥有了一个标准的 Windows 帮助环境。而 HLP 文件是通过帮助编译器编译后生成的,本文使用 HCW.EXE,为此需要准备以下文件:帮助文本文件(.rtf)、目录文件(.cnt)、项目文件(.hpi),下面分别说明。

1 HLP 文件的建立

1.1 帮助文本文件(.RTF)的建立

帮助文本文件是存放帮助说明及帮助主题的地方,它类似于超文本,点击文本中加亮的文字或图片,相应的帮助主题就会显示出来。帮助文本文件只能采用 RTF 格式,用微软公司 RTF 标准规定的符号集子集来说明文件中字符和段落的属性,它可以用任何一个支持 RTF 格式的编辑器进行编辑,本文使用 Word97,并结合奖学金管理系统帮助文件 JDXJ.HLP 进行说明。

(1) 建立帮助主题

帮助主题即帮助窗口中的信息,通常包含一个主题标题及一些列相关的信息。根据帮助信息的总体规划输入每一个主题,一个主题结束后插入一个分页符(用插入/分隔符/分页符功能)。如为 JDXJ.HLP 定义一个主题“奖学金输入”。

(2) 定义上下文标识符(Context String)

上下文标识符是帮助主题的唯一标识符,是帮助文件要跳转的目的标号,用脚注方式输入。方法为:将插入点移在要定义标识符的文字前面,选择“插入/脚注与尾注/脚注”,在“自定义标记”后输入“#”,选“确定”,在脚注位置输入标识符串即可。例如,我们为主题“奖学金输入”定义上下文标识符为“IDH_INPUT”。

(3) 定义标题、关键字列表和浏览顺序

与定义“上下文标识符”方法类似,相应的脚注为:

\$ 脚注:定义一个标题串。

K 脚注:定义关键字,关键字是帮助系统“搜索”时可以搜索到的一组字符串。用于搜索的关键字输入在 K 脚注中,关键字多于一个时用分号隔开。例如我们为主题“奖学金输入”定义关键字为“KEY_INPUT”。

+ 脚注:定义浏览顺序号,当用户浏览帮助文件时,帮助主题按定义的顺序显示。该顺序号实际上也是一个字符串,其顺序由该字符串在顺序号列表中所处的位置决定。

(4) 定义弹出式帮助文本

当用户点击弹出式帮助文本时,与其连接的帮助信息会出现在一个弹出式窗口。这样的文本在 HLP 文件中显示为绿色点划线文本,但在 RTF 文件中要定义为单划线,具体方法为:选择要定义的文本,加上单下划线,紧接其后(不能有其他任何内容)输入要跳转到的上下文标识符,并将该标识符去掉下划线,并标记为隐藏,方法为选中该标识符,从“格式/字体”中选中“隐藏文字”复选框。

(5) 定义跳转式帮助文本

当用户点击跳转式帮助文本时,与其连接的帮助信息会出现在当前帮助窗口(而不是一个弹出式窗口)。这样的文本在 HLP 文件中显示为绿色单下划线文本,但在 RTF 文件中要定义为双下划线,具体方法为:选择要定义

的文本,加上双下划线,紧接其后输入要跳转到的上下文标识符,并将该标识符去掉下划线,并标记为隐藏,方法同上。

(6) 插入位图

虽然 Word 支持图文混排,但帮助主题中不能直接存放位图,但可以通过引用命令语句来实现。方法为:在放置位图的位置输入:{bmc filename.bmp}。bmc 表示位图中放置,也可以用 bml 或 bmr 把位图放到左边或右边。若要将位图看作一个字符处理,则相应的命令为 bmcwd, bmlwd, bmrwd。

(7) 定义热点

热点是一个图形中可被激活的矩形区域,按上述方式放置的位图仅用于显示,并不能得到热点。如果要使位图得到热点进行跳转,需要用热点编辑器(Hotspot Editor)编辑位图。

用热点编辑器打开一个位图后,即可用鼠标定义一个热点区,热点区可以方便地被移动位置,改变大小,甚至删除。选择“Edit/Attributes”后得到一个 Attributes 窗口,其中 Context String:要跳转到的上下文标识符;Type:可以选择 Pop_up,Jump 和 Marco,说明该热点被点击时出现弹出窗口、跳转到别的主题上还是执行一个宏;其他项显而易见。你可以在同一个位图上定义多个热点,定义完毕后用 .SHG 存盘。将 SHG 位图文件插入到 RTF 文件中的方法同插入普通位图一样。

至此,一个图文并茂的帮助文本文件就建立好了。

1.2 目录文件 (.CNT) 的建立

利用 HCW.EXE 可以方便地建立扩展名为 .CNT 的目录文件,该文件类似于书的目录,定义了各章节的标题,就象您打开一个 HLP 文件时看到地那样。方法为:选择“File/New”,出现 New 窗口后选择“Help Contents”,单击“OK”,此时出现的窗口就是建立目录文件的地方。由于具体操作比较简单,在此不作详细,仅说明一点,若输入的条目是 Topic 时,对应的 Topic ID 就是 RTF 文件中的上下文标识符。

1.3 项目文件(HPJ)的建立

可以利用 HCW.EXE 方便地建立扩展名为 .HPJ 的项目文件,方法为:选择“File/New”,出现 New 窗口后选择“Help Project”,单击“OK”,然后依次设置 [Oprions], [Files], [Bitmaps], [Map], [Windows] ...等项,但并不是所有的项都是必须的。实际上,项目文件是一个标准的 ASCII 码文件,其结构类似于 .INI 文件,你可以用任何文本编辑器进行编辑,当然用 HCW 建立更直观。需要

说明的是, [Map] 选项将一个上下文标识符映射为一个数值,从而使帮助系统具有生成上下文相关的 (context-sensitive) 帮助系统的能力。例如,我们将上下文标识符“IDH_INPUT”映射为 100。

1.4 帮助文件 (.HLP) 的建立

利用 HCW.EXE 的编译功能方便地将帮助文本文件 (.rtf)、目录文件 (.cnt)、项目文件 (.hpj) 及其包含的位图等编译成 HLP 文件。

以上是利用帮助编译器 HCW.EXE 的方法, HCW.EXE 可以在 SDK, VC, Delphi 等环境中查找到。此外还可以利用一些专门的工具来建立 HLP 文件。

2 PB 中联机帮助的实现

PB 中提供了一个 ShowHelp 函数,利用它可以完成在线帮助,语法为:

ShowHelp (helpfile, helpcommand {, typeid })

其中:

helpfile: HLP 文件名;

helpcommand: 有三种取值:

Index! : 显示 HLP 文件中的顶层主题,该取值不需要第三个参数;

例如, ShowHelp (“JDXJ.HLP”, Index!)

Keyword!: 根据 HLP 文件中的关键字跳转,关键字由参数 typeid 指定;

例如, ShowHelp (“JDXJ.HLP”, Keyword! , “KEY_INPUT”)

Topic! : 根据 HLP 文件中的 MAP 的数值跳转,数值由参数 typeid 指定;

例如: ShowHelp (“JDXJ.HLP”, Topic! , 100)

这是最基本的语句,可以放在帮助菜单或帮助按钮的单击事件脚本中,但仅有这些是不够的,例如当用户按下 F1 键请求帮助时,还必须处理复杂的上下文相关帮助。

2.1 窗口内按钮的上下文相关帮助

窗口有一个叫做 KEY 的事件,每当在窗口的任何地方(输入焦点在 RichTextEdit 控件和数据窗口控件例外)按下一个键时,该事件被触发,因此可以在该事件中加入显示帮助信息的脚本。例如在奖学金输入窗口的 KEY 事件加入以下脚本,用来显示输入的帮助信息:

If Keydown(KeyF1!) then

ShowHelp (“JDXJ.HLP”, Topic! , 100)

End if

但是如果对窗口上不同的控件显示各自的帮助信

息,怎么办呢?要完成这个任务,KEY事件必须知道按键时焦点在哪个控件上,而该控件帮助信息的上下文标识符映射的数值可直接放在它的Tag属性中,在控件的得到焦点事件中加入下列脚本:

```
Parent.Tag = This.Tag
```

窗口的KEY事件的脚本也要改为:

```
.....
```

```
ShowHelp ("JDXJ.HLP", Topic! ,  
Long(This.Tag) )
```

2.2 数据窗口的上下文相关帮助

当焦点处于数据窗口上时,不触发窗口的Key事件,而数据窗口本身也没有Key事件,不过我们可以创建一个新的事件。单击数据窗口,选择“Declare/User...”,显示Event对话框。在位于窗口底部的Event Name空白字段处输入Dwkey,在对应的Event ID列的空白处输入pbm_dwnkey,然后选择OK。实际上,并没有创建一个新的事件,只是用来捕获已经存在的KEY事件。在现在为数据窗口编写脚本,选择事件Dwkey,加入与窗口的KEY事件相同的代码:

```
If Keydown(KeyF1!) then
```

```
ShowHelp ("JDXJ.HLP", Topic! , 100)
```

```
End if
```

这种做法对于简单的帮助系统是可以的,但是在多数情况下,数据窗口将含有更多的信息,如果一个复杂的应用程序对于整个数据窗口仅有一个帮助主题,就不能说

这个帮助系统是非常上下文相关的。因此,怎样用不同的基于列的上下文相关帮助呢?答案是使用函数GetColumnName(),事件Dwkey的代码也要改变如下:

```
String Colm_name
```

```
If Keydown(KeyF1!) then
```

```
Colm_name = This.GetColumnName()
```

```
Choose Case Colm_name
```

```
Case "ColumnNam_1"
```

```
ShowHelp ("JDXJ.HLP", Topic! , 100)
```

```
Case "ColumnNam_2"
```

```
ShowHelp ("JDXJ.HLP", Topic! , 200)
```

```
.....
```

```
Case Else
```

```
ShowHelp ("JDXJ.HLP", Topic! , 100)
```

```
End Choose
```

```
End if
```

用数据窗口中的真实列名替换Case语句中的列名,用实际值来替换ShowHelp的参数,在省略号(.....)的地方可任意增加需要的语句。■

参考文献

1 Bill Hatfield. *PowerBuilder 5 应用程序开发指南*. 北京: 清华大学出版社, 1997

2 樊金生, 傅世林, 邱书灵等. *PowerBuilder 6.5 使用教程*. 北京: 科学出版社, 2000