FoxPro for Windows 使用技巧(连载三)

罗　辉　　(湖南省双峰工商银行)

54.报表变量的使用

报表变量允许你在输出报表的每一条记录时作一些计算,其计算结果可为尔后其他的计算引用,也可象其他变量一样在报表中使用报表变量。譬如你要作一个时间安排表,在表中将用到这些数字字段:hour_in、min_in、hour_out、min_out。你可定义三个报表变量:

ARRIVE = HOUR_IN + (MIN_IN / 60)

LEAVE = HOUR_OUT + (MIN_OUT / 60)

DAYTOTAL = LEAVE - ARRIVE

其中第三个变量引用前面两个变量的结果,并在报表中使用。

使用报表变量必须注意:

①报表变量初始化的顺序很重要。譬如,如果变量 var1 被用来定义变量 var2,那么在 Report Variables 对话框的报表变量列表中,var1 必须先于 var2 被列出,以便它首先被计算。

②报表变量什么时候复位也很重要。一般地,有四个复位的位置:报表结束、页结束、(蛇行)栏结束、组结束。默认时,报表变量将到报表打印完后被复位到初始值。

③报表变量可用来对数据库中的记录数据进行计算。例如,要统计某一城市(加利福尼亚)中公司的个数,在报表变量定义 (Variable Definition)对话框中,你先建立一个变量,它赋值为 IIF(state="CA",1,0),然后选择 Sum 计算即可。

55.满页打印和有效区域打印

在 Page Layout 对话框中,可以选择报表的满页(Whole Page)或有效区域(Printable Page)打印方式。

选择有效区域打印方式,FoxPro 将按照默认打印机的边界定义对报表的边界进行设置。报表有效区域外的信息将不会被裁剪,如果报表的左边界为 0,将使报表内容紧挨有效区域的边界打印。

如果你的报表将在一组打印机上打印,则选择满页打印,它允许你更可靠地控制报表的打印,而不依赖于具体是哪种打印机。如果报表的左边界为 0,则报表内容将紧挨打印纸边界打印。你可能要在 Report Layout 窗口给报表边界保留一定的空白,以便在各种打印机下都能将所有报表数据打印出来。

报表预览,将显示在当前打印机下被打印出来的报表的真实外观。

56.强化的 RUN 命令

RUN 命令提供一个新的运行参数 /N,以便在 FoxPro for Windows 环境直接调用 Windows 应用程序。

例如:RUN /N C:\WORD\WINWORD.EXE README.DOC 将调用 Word for Windows 软件编辑一个叫 README.DOC 的程序。

Windows 的控制板内带有许多实用程序,你可用 RUN/N 命令在你的应用程序中直接调用它们,如:

RUN /N CONTROL COLOR　　&& 调用色彩拾取器

RUN /N CONTROL PRINTERS && 调用打印机管理器

RUN /N CONTROL DESKTOP　&& 调用桌面管理器

用 RUN /N 或 ! /N 命令执行一个 Windows 应用程序,与在程序管理器或文件管理器中执行程序其内部管理方式是一样的,因而你可以进行诸如在该应用程序和 FoxPro for Windows 之间切换任务的标准 Windows 操作。

你可以在参数 /N 后再附加一个可选项数值(中间不能含有空格),以指定该应用程序的运行方式。

该可选项参数值及其意义如下表:

/N 可选项参数值　应用程序运行方式

1　激活应用程序窗口并以正常窗口大小运行

2　激活应用程序窗口并以窗口最小化运行

3　激活应用程序窗口并以窗口最大化运行

4　以正常窗口大小运行,但非活化应用程序窗口

7　以正常窗口最小化运行,但非活化应用程序窗口

57.驱动器、目录和路径信息的获取

有几个 FoxPro 函数用以获得有关驱动器和目录的信息。下面列举有关函数的使用实例,注意比较使用 SET DEFAULT TO C:\FOXPROW 命令后的返回值的不同。

获取驱动器信息的函数:

SYS(5) = C:

SET('DEFAULT') = C:

获取目录路径信息的函数:

CURDIR() = ¥FOXPROW¥

SYS(2003) = ¥FOXPROW

FULL(SET('DEFAULT')) = C:¥FOXPROW¥

如果要判断某一指定的路径是否存在,你可以用 ADIR( ) 函数来测试:如果指定的 C:¥FOXPROW¥ 路径不存在,则 ADIR(temparr,'C:¥FOXPROW¥*.*','D') 函数返回 0 值。注意:如果是测试网络路径,若网络联结被打断,则指向网络服务器的路径将不存在,ADIR( ) 函数会返回 0 值。

58.强化的 USE 命令

在 FoxPro 环境,USE 命令能力被大大强化。如使用 USE <数据库名> IN 0 命令可以在当前未被利用的最小的工作区打开一个数据库文件,它使你在打开数据库之前不必先用 SELECT 命令选择该工作区;同样,USE IN <别名> 命令关闭该别名指定的工作区中打开的数据库文件。

USE 命令还提供了一条 SHARE 子句以允许你在没有执行 SET EXCLUSIVE 命令时以共享方式打开一个数据库。

59.在状态栏显示用户信息

一般情况下,FoxPro 状态栏显示系统的有关信息。如果

你希望在状态栏显示用户的特定信息，那么，首先执行 SET TALK OFF 命令，然后用 SET MESSAGE TO <信息字符串> 命令指定要在状态栏显示的用户信息即可。

执行不带参数的 SET MESSAGE TO 命令，将恢复正常显示。

60.不要用预处理指示器定义有关跨平台的系统变量

不要将 _WINDOWS、_DOS、_MAC 或 _UNIX 等系统变量与预处理指示器（如 #IF...#ENDIF）混合使用，因为这些系统变量是一些实时变量，只有在实时运行时才有定义，比如下面的程序代码：

```
#define DOS .t.
#define -WINDOWS .f.
DO CASE
CASE DOS
? "程序在 MS-DOS 下运行"
CASE WINDOWS      && 在 DOS 下编译，则在 Windows 下不会执行如下语句
? "程序在 Windows 下运行"
ENDCASE
```

如果上述程序代码在 MS-DOS 下编译，则在 Windows 平台将不能正常执行。因为在编译时，_DOS 变量就被赋值为 .T.，而 WINDOWS 被赋值为 .F.，以致代码永远也不能被正常执行。如果用下面的代码取而代之，就可避免上述错误：

```
#if "Win" $ VERS()
#define WINDOWS_CODE .t.
#elif "Mac" $ VERS()
#define MAC_CODE .t.
#elif "Unix" $ VERS()
#define UNIX_CODE .t.
#else
#define DOS_CODE .t.
#endif
#if WINDOWS_CODE
? "代码在 WINDOWS 下编译"
#elif DOS_CODE
? "代码在 DOS 下编译"
#endif
```

61.防止预处理器置换文本

预处理器 #DEFINE 用于定义编译常量名，在编译时，程序代码中凡出现这些编译常量名的地方都将用该常量名对应的值进行替换。但可能程序中某处文本中出现了与编译常量名相同的字符串，而你又不希望被对应的常量值替换，则应用方括号 "[]"将该字符串括起来。但应注意的是：在将数组下标变量与编译常量名混用时，将引起错误。譬如：

```
#DEFINE nosub 1
DECLARE an_array(3)
? afunc[an_array[nosub]]
FUNCTION afunc
PARAMETER x
RETURN x
```

上面的程序在运行到 ? afunc[an-array[nosub]] 语句时，将会出现"变量 NOSUB 没发现"的错误。因为编译该程序时，下标变量 nosub 没有用编译常量值进行替换，待到运行时又因没有被定义而出错。

62.使用增强的开发工具

FoxPro for Windows 根目录中的 FOXTOOLS.FLL 文件中包含了一组函数，它们可用来增强你的应用能力。这些函数大部分提供存取 API 例程和 Microsoft Windows 函数的能力。你可以在用 SET LIBRARY TO 命令指定该文件为活动的库文件后，用 DISPLAY STATUS 命令查看这些增强的函数（内含 114 条函数）。有关这些函数的使用，请参考相应的开发指南。

63.跨平台应用不兼容代码的处理

当设计一个跨平台的应用时，有许多代码可以透明地跨平台使用（如数据库操作的代码），但另有一些则不能跨平台执行（譬如一些应用程序接口函数），因此必须针对不同平台将这些特定的代码分别处理。一般地，可使用 IF...ENDIF 和 DO CASE...ENDCASE 的结构，结合诸如 _DOS、_WINDOWS、_MACINTOSH 和 _UNIX 等系统内存变量，将多个平台不能混用的代码分开处理，以保证不同代码只在相应平台才被执行。

64.跨平台应用修改的不灵敏性

你可以透明地在各个不同的平台上运行 FoxPro 2.5 应用程序，也可以在一个平台对应用程序进行修改，而不必在所有平台上都对应用作同样的修改。但必须注意，无论如何，你在一个平台所作的改变将不会自动地反应到另一个平台上去。譬如，在 FoxPro for Windows 中，在屏幕上添加一个对象并重新编译这个屏幕程序后，如果在 FoxPro for MS-DOS 平台运行这个 .APP 应用，则新添的对象将不会出现在屏幕上。你必须在 FoxPro for MS-DOS 平台打开这个屏幕并重新编译这个应用。它将弹出一个 Transporter 对话框，显示这个新对象在另一个平台上添加了进去，就象是在该平台直接添加了该对象一样。只有这样处理，才能使用该对象。

65.跨平台转换屏幕元素

当跨平台转换应用程序时，有些屏幕设计可能需要调整，特别是当你从一个字符平台转换到一个图形平台时更应如此。如果你打算将 Windows 应用程序转换到 MS-DOS 下，则应保留更多的屏幕空间，因为在图形环境的屏幕对象是精确到象素，而字符环境是精确到行列。

另外必须注意的是，应使用两个平台都有效的屏幕对象，诸如 OLE 对象和图象对象在 FoxPro for MS-DOS 下是不能发挥作用的。　　（全文完）