

# 第一篇 基础知识

# 第一篇 基础知识

# 第一章 重要命令及其运用技巧

## 第一节 阅读本书的技巧

本书的重点是在程序设计部分,也就是在第二篇的录像带管理系统以及第三篇的补习班管理系统,录像带管理系统中我们以稍微简单一点的设计方法来设计,所使用的 FoxPro 命令也比第三篇中补习班管理系统中的少,因此我们可以循序渐进地了解 FoxPro 的程序设计技巧。

在第一篇中,我们将介绍本书中所使用的重要概念以及重要命令的运用,我们之所以要将重要的命令放在这一篇中,是为没有工具书的读者准备的,以便于读者在不懂本书中所使用的命令与函数时,可在这一篇中得到您所想要的重要数据,在此顺便介绍了程序设计的重要概念及技巧。

在第二篇中,我们以写作方式设计了一套完整的软件“录像带管理系统”,我们之所以以“录像带管理系统”为例,是因为在日常生活中,计算机化最明显的商店是录像带店,因此我们以这一系统为例来介绍程序设计技巧。

在第三篇中介绍补习班管理系统,以电脑补习班为例是由于社会上各种补习班比较多,而且补习班的管理又易于电脑化。

### 第一篇的使用说明

1. 在命令行中[]表示是可选择的,是可有可无。
2. /表示前项或后项只能选择其一,如 On/Off
3. 缺省值用大写字母表示,如 SET EXACT on/OFF
4. (Exp C) 表示字符型表达式  
(Exp N) 表示数值型表达式  
(Exp D) 表示日期型表达式  
(Exp L) 表示逻辑型表达式  
(Exp R) 表示任何类型的表达式
5. 命令行中的所有字符大小写均可。
6. (ExpC1)、(ExpC2)、…、或(ExpN)、(ExpN1)、(ExpN2)表示有其中一个以上的表达式或信息。

## 第二节 关于界面

? 及??

语法

? [**<Expr1>**][**Picture<ExpC1>**][**Funtion<ExpC2>**][**AT<ExpN>**][**Expr2**]

说明

(1)<Expr1>

第一个被? 或?? 使用的任一类型表达式。

(2) **Picture <ExpC1>**

? /?? 将以<ExpC1>的字符串表达式为样板格式输出,其中样板将在后面介绍。

(3)**Funtion <ExpC2>**

使用功能代号作格式化输出(稍后介绍)。

(4)**AT<ExpN>**

指定从那里开始输出。

(5)<Expr2>

第2个,第3个,...被? /?? 以使用的任一类型的表达式或信息。

范例

? "Chinese is good" Function "!" ←以大写形式输出

? Date() Function "E" ←以欧洲日期形式输出

ID="f100200300"

? ID Picture "I 999999999" AT 25

根据上面的范例可以了解到以下几点:

(1)??表示接上一行连续输出,?是在print之前先打印一个CHR(10),CHR(13),因而要跳到下一行。

(2)若将Printer 设置为ON,则? /?? 将输出到打印机,但@...SAY 不受影响。

@...SAY | GET

语法

@(<行,列>)[**SAY<Expr1>**]

[**Picture <ExpC1>**]

[**Function <ExpC2>**]

[**Color Scheme <ExpN1>/Co**

[**Color Scheme <ExpN1>/Color**

[**Color Scheme <ExpN1>/Color<Color—Pair>**]

[**GET <变量>**]

[Picture <ExpC3>]  
 [Function <ExpC3>]  
 [Default [Expr2]  
 [Fenable/Disable]  
 [Range[<Expr3>][[,<Expr4>]]  
 [Size <ExpN2>,<ExpN3>]  
 [Message <ExpC5>]  
 [[Open] Windows <Window—name>]  
 [Valid <ExpL1>/<ExpN4>[Error<ExpC6>]]  
 [When <ExpL2>]  
 [Color Scheme <ExpN5>/Color <Color—Pair>]

#### 说明

##### (1) <行,列>

是指定屏幕上的行列位置,但因显示器不同而范围也可能有所不同,如 SET display TO VGA50 则行可设置到 49。

(2) SAY 在<行,列>位置输出(Expr1)的内容,当然若您使用了 Picture 或 Function 的话, SAY 自然会根据其格式来输出。

##### (3) Picture 的样板符号:

A→表示只能使用英文字母

N→表示只能输入英文字母或数字

X→表任意字

L→表示只可输入逻辑类型的数据

9→只可输入 0—9 或 +、—号

Y→表示只接受 Y 或 y、N 或 n,并转化成大写

\$→用来显示货币符号

.→用来指定小数点位置

,→以三位一逗号方式显示

!→将小写转化大写(只对英文字母有效)

#→只接受 0—9,空格及 +、—

\*→在数值数据之前打上\*号,通常与\$联合使用

##### (4) Function 的功能符号

A→只接受英文字母

(N)B→将数值数据左齐输出

(N)C→在输出的数值后加上‘CR’符号(只能用于 SAY 语句)

D→将数据以当前 Date 的设置格式输出(SET DATE 后述)

E→将数据按欧洲日期格式输出

J→将数据右齐输出

I→将数据列于输出字段的中间

(N)L→当数值类型的数据前有空格时补为零

(C)R→非样板符号可显示,但不在变量之中如 Picture " (999) 999—9999" 其中的

( ) 及一号便是一例

T→去掉字段前后空格

M(List)→(List)中可放置一些可选的选项,其间可用逗号分开

(C)(N)S→此功能符号可用来设置字段宽度,但不改变变量或字段的实际长度,可用左、右箭头的按键来实现字段的水平滚动

(N)X→表示数值数据若为负,输出之后将会显示'DB',且只能用于 SAY,请参考 C 的功能符号。

(N)(→表示数值数据若为负数则加上(),但只用于 SAY。

(C)!→所输入的任何字符中,英文字母会转成大写,其他字符不变。

(N)^→将数值数据以科学表达式形式输出。

(N)Z→当变量值为 0 时,则以空格输出。

(N)\$→数值数据将以货币的格式输出,即右前或右后加上\$符。

(5)Color Scheme (ExpN1)/Color (Color Pair)

Color Scheme (ExpN1)将以所指定的颜色显示 SAY 之后的表达式或信息,其范围为 1—24。

而 Color (Color—Pair)则是设置颜色,至于颜色代号请参见下一章。

注意:在功能符前的(N),(C)等表示只有在 N,C 类型数据才能使用。

范例

```
ID—"E201111303"
```

```
@:1,1 SAY ID Picture " ! 999999999" ←结果 E201111303
```

```
@:1,1 SAY 12000 Picture "999,999"
```

```
GET (变量)
```

一个好的应用软件必须有良好的人机接口,让用户能方便地输入数据,在制作界面之中以 Get 命令最为重要,如建立一个字段让用户输入数据或是建立一个 CheckBox 或 Push Button 等都是使用 Get 命令来完成的,一般输入界面如下:

```
@:5,1 SAY "请输入姓名:"Get Name
```

```
Read
```

相信这种界面方式是读者所熟知的,但是其缺点甚多,现随便举一二个例子来说明。如果用户应输入的是姓名,结果却错误地输入一串数字或只键入一个姓,请问该如何,不外乎 Read 之后再检查,再跳回 Get 该行操作,这或许是个好方法,那我们再举个例子:

```
@:4,1 SAY "学号:"Get S_No Picture "9999"
```

```
@:5,1 SAY "姓名:"Get Name
```

```
Read
```

在这个例子中是要输入学号与姓名,但是若用户在学号处没有键入数据就要输入姓名,

相信这在往后的处理中必出问题,如建立索引或按学号查找数据时就会出错。那么该如何得知用户到底有无输入学号,才来决定是否允许输入姓名或根本不允许用户离开该 Get,这在 FoxPro 中都有很好的解决方法,让我们一起来了解这一含有强大功能的命令吧!

(1)Picture 及 Function 二参数前面已说明,这就不再重述了。

(2)Default;

一般我们在使用一个变量之前要先对定义,或赋予一值,如果未定义而直接使用,则会出错,使用 Default 参数可允许您 Get 一个未经定义的变量,如:

```
...Get Temp Default "A001"
```

此用法相当于

```
Temp="A001"
```

```
...Get Temp
```

聪明的您可能马上会问,如果事先已定义,在使用时又加了 Default,则会怎样? 这时 FoxPro 就以先前定义的主而忽略了 Default。

(3)Enable/Disable

这两个参数是让该 Get 使用或禁止使用,若用户某步骤未完成而影响该 Get 的运行,则可将该 Get disable,就如同在 FoxPro 中没有 use 文件时不能使用 Record。

(4)Range[(expr3)][,(Expr4)]

顾名思义,Range 参数是用来设置范围的上下限,而上下限的类型必须和 Get 的变量相同,当然可以只设置上限或下限,若只限制上限或下限,则只检查一边。

(5)Size (ExpN2),(ExpN3)

此参数的功能在于改变 Get 的编辑字段大小,缺省为一行长。

(6)Message (ExpC5)

在 FoxPro 中每一个 Get 对象(Object)几乎都可以带说明信息,并显示在屏幕的最下行。

(7)[[OPEN]Window (Window--name)]

本参数可允许自行开窗口来编辑 Memo 字段。

(8)Valid

这是一个很重要的参数,通常我们用它来检查该 Get 所输入的数据或其所生成的事件(Event)如 Push Button 时即产生另一个事件,(ExpL1)通常是一个用户自定义函数,而该(UDF)而返回.T.或.F.,举个例子:

```
@...SAY "输入学号" Get S No Valid MyFu(S.No)
```

```
...SAY "输入姓名" Get Name
```

```
Read
```

```
Function MyFu
```

```
Parameter CHK
```

```
IF Empty (CHK)
```

```
Wait "此字段不能为空" Window Nowait
```

```
Return .F.
```

```
ENDIF
```

Return . T

而(ExpN4)是对数值表达式检查时,Valid 参数会返回一个数值,告诉 FoxPro 下一个字段是第几个 Get 会被激活,Valid 会返回的数值有 0、正负值,0 表示输出有错,将不离开该 Get,负值表示将退到前 N 个 Get 去 Read,若为 +N 则表示前进 N 个 Get 去 Read。

(9)When (ExpL2)

此参数通常用来检验该 Get 被 Read 之前先判断(ExpL2)是否为 . T. ,若为真,则进行输入否则不能输入,再以前例来说明:

学号不输入则姓名不能被 Read

.. SAY "输入学号" Get S\_No Valid MyFu (S\_No)

.. SAY "输入姓名" Get Name When not Empty (S\_No)

Read

(10)Color Scheme (ExpN5)/Color (Color—Pair)

如前所述,此参数可以自行使用第 1 至第 24 个颜色盘来定义颜色或使用颜色代码。

注意:Picture 及 Function 的功能符号有冲突者不可混用,如"@DE"。

## @..Get (检验框 Check Box)

语法

@(行,列) Get (变量)

Picture (ExpC1)/Function (ExpC1)

[Enable/Disable]

[Default (Expr)]

[Message (ExpC2)]

[When (ExpL1)]

[Valid (ExpL2)]

[Color Scheme (ExpN1)/Color (color—pair)]

[Size (ExpN2),(ExpN3)]

熟悉 Windows 的读者对 Check Box 一定不陌生,有一个选项中出现如下所示的一行消息:

save when exit

然后可在该消息处按一下鼠标,就打开了该选项,这就是我们这儿所说的 Check Box。在一般软件中,要制作这项功能可能很费力,但在 FoxPro 中,我们只要使用一行命令即可完成这一功能。

说明

(1)(行,列)

指在屏幕上所定的位置

(2)Get(变量)



该式中的变量可以是一内存变量, FoxPro 的变量, 或一字段, 若该变量值不为 0 或逻辑变量为 .T., 则 ☐ 会自动打上 ×, 表示已打开, 反之则 ☐ 内不会打 ×。

(3) Picture (ExpC1)/Function (ExpC2)

此处的样板符及功能符号与 @... SAY 的符号不太一样, 它有自己的符号即为 \*C。

\*C: 为 Check Box 的符号, 若在 Picture 中使用, 请在前方加 @, 若在 Function 中使用, 则免。

N: 表检查完该 Check Box 并不立即离开, 而可继续检查其他 Box, N 为缺省值。

T: 恰好与 N 相反。

例: @1, 1 Get All -- F Function " \*C \<A 查找所有数据"

@2, 1 Get Cases Function " \*C \<C 不论大小写"

@3, 1 Get Leave Picture "@ \*C \<Q 查找后结束"

此例即产生三个 Check Box (请留意变量假设值已设置, 以及在双引号内的 \<, \, " \<" 表示其后的字符为热键 (Hot key), 只要按下即对该检验框作选择, 并注意一点, Check Box 是以 0, 1, ON/OFF 的概念来做的。而 "\<" 表示不可选的。)

(4) Default

如前所述, 在变量未定义之前使用它是会出错的, 但在 Default 参数后, 则可排除这个错误, 但若变量已定义则将会忽略 Default 值, 如下例:

未声明:

@1, 1 Get ok " \<To Screen" default . T.

声明:

OK. =. F.

@1, 1 Get ok " \< To Screen " Default . T.

则值仍为 . F.

(5) Enable/Disable

此参数最主要的功能是激活或禁止该 Check Box, 通常我们会先作一些必要的检查步骤, 再 disable 某个 Check Box。

(6) Message (ExpC2)

此参数中的 ExpC2 是用来说明 Check Box 的消息, 缺省位置为屏幕的最后一行。可用 SET MESSAGE 来设置位置。

(7) When (ExpL1)

我们可以这样说, When 条件是进行 Read 前的检查, 因此只要 (ExpL1) 为 . F., 则此 Check Box 就不能进行 Read, 反之若条件为 . T., 则可以。且 (ExpL1) 可以是用户自定义函数, 由 UDF 自己进行一些比较复杂的检查工作。

(8) Valid (ExpL2)

此处的 Valid 与前面所说的 @... Get <变量> 有些不同, 这里的 VALID 常配合自己的 UDF 使用, 如下列:

store . T. To S, T

@1, 1 Get s Function " \*C \<Search All"

```

@2,1 Get T Function " * C\ \Quit When find";
Valid My(T)
:
Read Cycle    ← 会在这两个 check Box 不断 Read 直到按 ESC
Function My
Parameter CHK
IF! CHK
    Wait"      ← 此检验框最好选择 ON" window
    Return .F.
ELSE

```

:(此处处理.T. 的状况)

ENDIF(通常为终止 Read 命令,即用 Clear Read)

(9)Color Scheme (ExpN1)/Color (color—pair)

此参数允许用户自行使用 FoxPro 所缺省的颜色盘,值为 1—24。若不使用 color scheme 的颜色则可使用自行定义的颜色。

@...color,,,B/W,Gr+/B,R/Gr,B/W,R/W,或许对于此处连续 4 个",表示疑问,原因是 color—pair 有十组颜色,现将其说明如下

第 5:Message 的(ExpC2)颜色

第 6,当前所在检测框的颜色

第 7:热键颜色

第 9:Enable 的检测框颜色

第 10:Disable 的检测框颜色

其他未列的则不被 Check Box 所使用,故前 4 个为空白。

注意:没有使用的要用', '隔开。

(10)Size (ExpN2),(ExpN3)

此参数设置了检测框的高度(ExpN2)及宽度(ExpN3)

@...Get 列表(List)

语法

@(行,列) GET (变量) /字段

[Function (ExpC1)/Picture (ExpC2)]

[From (数组)]

[Range (ExpN1)[,(ExpN2)]]/Popup (popup..Name)

[Default (ExpR)]

[Size (expN3),(ExpN4)]

[Enable/Disable]

[Message (ExpC3)]

```
[Valid <ExpL1>]  
[When <ExpL2>]  
[Color scheme <ExpN5>/Color <color pair>]
```

#### 说明

首先说明的是@...Get(List)产生一个列表,通常是一个弹出式下拉式列表,但并非绝对如此,可根据需要而定,现先来了解一下参数的内容。

##### (1)<行,列>

用来定义屏幕上的坐标位置,当在 VGA50 模式下行可定义到 49。

##### (2)Get <变量>/字段

此处变量可以是内存变量,也可以是数据库中的字段,但要注意的是其数据类型必须是数值型或字符型。

##### (3)Function <ExpC1>/Picture <ExpC2>

此处 Get 没有前面的“\*1”或“\*C”标识符,而只有:

&T:表示用户选其中某一项即停止。

&N:表示用户选择某项之后不停止 Read 命令。

##### (4)Range <ExpN1>[,<ExpN2>]

指定从数组中第<ExpN1>个元素开始,<ExpN2>是用来指定要写进几个元素。

Popup <popup\_Name>;建立一个列表,而列表内容是由 Define Popup 命令来定义的。

##### (5)Default

如前所述,若使用变量未事先定义则可使用 Default 命令参数。

##### (6)From<数组>

建立一个列表通常是按一个数组的内容来建立的,而此数组可以是一维或二维数组,若为一维数组则第一个元素内容会放入该 List 的第一个选项,若二维数组则第一栏的第一个元素会放入 List 中的第一选项,第一栏的第二个元素则放入第二个选项,依此类推!

##### (7)Size <ExpN3>,<ExpN4>

FoxPro 可以很容易地根据<Exp3>来改变它的高度,或按<ExpN4>来更改 List 的长度。

##### (8)Enable/Disable

这两个命令行参数用来打开或关闭该 Get 能否被 Read 使用。

##### (9)Message <ExpC>

用来显示该 List 的信息,而此信息是根据<ExpC3>的值来显示。

##### (10)Valid <ExpL1>

通常会我们会根据 Valid 值来判断是否要离开该 List,而此命令参数又常配合用户自定义函数(UDF)做许多较复杂的工作,实际操作请参见后两篇。

##### (11)When <ExpL2>

在要对@...Get 进行 Read 时,FoxPro 会先检查 When 之后的<ExpL2>是否为 T.,若为 F. 则跳过该 Get。

##### (12)Color Scheme <ExpN5>/Color <color pair>

color scheme 可以让用户使用第 1 至第 24 个颜色盘,或利用 color <pair>自行设置颜色。

范例

请参考补习班系统中的文件编辑。

#### @...GET (隐式按钮 Invisible Button)

语法

@ (行,列) GET (变量)

Function (ExpC1) /Picture (ExpC2)

[Default (ExpN1)]

[Size (ExpN2), (ExpN3) [, (ExpN4)]]

[Enable/Disable]

[Message (ExpC3)]

[Valid (ExpL1)]

[When (ExpL2)]

[Color Scheme (ExpN5) /Color (color\_pair)]

说明

所谓隐式按钮是按扭本身并未提供提示符(Prompt),而是自己先将按钮本身所代表的字符或图案画在屏幕上,再将这些按钮显示在提示符的地方,让大家以为这也是选项(事实上本来就是)。

参数说明如下:

(1) (行,列)

指定屏幕上的坐标位置,若使用 VGA50 模式则可达 49 行。

(2) Get(变量)

<变量>可以是内存变量或 FoxPro 的内定变量,或数组元素,数据库字段等。

(3) Function (ExpC1) /Picture (ExpC2)

① \* 1:此符号是用来识别这个 Get 是什么类型的 Get,1 表示隐式(Invisible)。若要建立一个以上的按钮请用“;”,将其分开。

② N:表示当按 Enter 或单击鼠标左键选择 Button 时,并不立刻停止 Read 的功能而可以继续选择别的 Button,此为缺省值。若不停止 Read,则必掉入这一死循环中,因此后面的 Valid 就显得非常重要。

③ T:表示选择按钮后即中止 Read 命令。

④ H:表示要用水平方式,即横向方式排列这 Button。

⑤ V:表示用竖式排列。

以上代号可混用如 NV, NH, TV, TH, 但相冲突者不能混用,如 NT, HV。

另外,要设某个按钮为不可选,则在前加上“\”。

#### (4)Default <ExpN1>

若变量事先未定义,则可用此命令参数使之自行赋值定义。

#### (5)Size <ExpN2>,<ExpN3>,[,<ExpN4>]

此命令参数用来设置按钮的大小,FoxPro 会根据<ExpN2>来更改高度,<ExpN3>更改长度,<ExpN4>则是两个按钮间之间距,此间隔距离单位为 Byte,即字节。

缺省的高度及长度为 0。

#### (6)Enable/Disable

此命令参数用来打开或关闭一个 Get 以被 Read 命令读取。

#### (7)Message <ExpC>

此参数将以<ExpC3>的内容作为 Get 的提示信息。

#### (8)Valid <ExpL1>

通常在选择一个 Button 之后即利用 Valid 进行检查工作,如前所述,若使用功能符号“N”及 Read Cycle 时不就无法离开了吗?因此我们可以用 Valid 来实现离开该 Read 的操作,Valid 后常接 UDF。

#### (9)When <ExpL2>

在对此 Get 进行 Read 动作时,FoxPro 会先对 When<ExpL2>作检查,若<ExpL2>为.F.,则不进行该 Get,反之则激活该 Get。

#### (10)Color Scheme <ExpN5>/Color <color-pair>

Color Scheme 可以让用户自行使用颜色盘,范围值为 1-24。

Color <color pair>即为让自行设置的顏色。

#### 范例

##### (1)OK=1

```
@1,1 Get ok Function " * I ; ; " SIZE 2,2,3
```

Read Cycle

建立 4 个按钮

##### (2)OK=1

```
@1,1 Get ok Picture " * I N ; ; " Size 2,2,4
```

Read Cycle

建立三个一直重复的按钮

##### (3)@1,1 Getok Function: \* IV ; ; " Size 2,2,3:

Default 2

Read Cycle

建立四个竖式按钮,初始值为 2,按钮大小为 2\*2,间距为 3。

@...Get (弹出式菜单 . Popup)

## 语法

@<行,列> GET <变量>

Function <ExpC1>/Picture <ExpC2>

[Default <Expr1>]

[From <Array>]

[Range <ExpN>[,<ExpN2>]

[Size <ExpN3>[,<ExpN4>]]

[Enable/Disable]

[Message <ExpC3>]

[Valid <ExpL1>]

[When <ExpL2>]

[Color Scheme <ExpN5>/Color <color\_pair>]

## 说明

弹出式菜单较一般菜单不同的是,其选项可能有好几个,但是只在屏幕上显示一个选项,在要选择该菜单内容时弹出一个窗口并将所有选项列在其中,可用↑↓等键来选择其中的选项。

命令参数说明如下:

(1)<行,列>

定义屏幕上的坐标位置。

(2)Get<变量>

此变量可为内存变量或数组或字段,且只能为数值或字符类型,因此若变量为数值时,则会传该个选项在这菜单中的顺序,如果变量是字符类型则传给该变量此选项之提示字符。

(3)Function <ExpC1>/Picture <ExpC2>

①^:此符号就好比 Check Box 的 \*C-一样是个标识符。

②N:选择选项后不立刻停止 Read。

③T:恰好与 N 符号相反,选择选项后会立刻停止 Read。

(4)Default <Expr>

当使用的变量未事先定义时,则可使用 Default 命令参数来设置。

(5)From <Array>

菜单中的选项即来自此数组的元素。

(6)Range <ExpN1>[,<ExpN2>]

<ExpN1>是指由数组第几个元素开始,取<ExpN2>个元素值。

(7)Size <ExpN3>[,<ExpN4>]

<ExpN3>可更改 popup 的高度,而<ExpN4>则用来更改宽度。

(8)Enable/Disable

此两个参数则用来允许或禁止 Read 功能。

(9)Message (ExpC3)

此即为 popup 的提示信息。

(10)Valid (ExpL1)

Valid 一般使用时机是当用户已选择选项后进行判断工作,通常后接 UDF。

(11)When (ExpL2)

Valid 是作事后判断,而 When 则作事前判断,目的在于是否允许 Read 命令去激活它。若 (ExpL2) 为 True,则可运行,反之则否。

(12)Color Scheme (ExpN5)/Color (color pair)

Color Scheme (ExpN5)可允许用户指定颜色编号,而 color (colorpair)则表示由用户自行设置颜色。

范例

@1,1 Get ok Function "^\Car;\Motorbike;\Bike "Default 2 Size 2.15 在这里看到的  
\<及\,\表示其后第一个字符为热键,\表示该项不能选择。

Read Cycle

@...Get <按钮 Push Button>

语法

@<行,列> GET <变量>

Function (ExpC1)/Picture (ExpC2)

[Default (Expr)]

[Size (ExpN1), (ExpN2) [, (ExpN3)]]

[Enable/Disable]

[Message (ExpC3)]

[Valid (ExpL1)]

[When (ExpL2)]

[Color Scheme (ExpN4)/Color (color. pair)]

说明

这种按钮在很多软件中都见得到,最常见的恐怕就是这种。

<Ok>,<Quit>或<Open>,<Close>等命令参数与前面几个大致相同,我们只作简单介绍。

(1)<行,列>

定义坐标位置。

(2)Get<变量>

此变量是用来存放所选择的按钮顺序编号,因此完全由此变量的类型而定。

(3)Function (ExpC1)/Picture (ExpC2)

①\*:这儿的\*号表示是 Push Button,若超过一个 Push Button,则用“;”来分开,通常也

是超过一个。

②N:当选了某个 Button 之后,不会立刻跳离 Read,而一直继续下去。

③T:恰好与 N 相反,选了某个 Button 之后,则立即跳离。

④H:表示要建立一个横向的 Push Button。

⑤V:表示要建立一个垂直的 Push Button。

(4)Default (Expr)

当未事先定义(变量)时,可以用 Default 参数来自行建立,且可缺省激活按钮为第几个,甚为方便。

(5)Size (ExpN1), (ExpN2)[, (ExpN3)]

〈ExpN1〉可用来设置高度,〈ExpN2〉可用来设置宽度,而〈ExpN3〉则设置间隔,单位为 Byte。

(6)Enable/Disable

此参数用来允许或禁止 Read 能不能来激活此 Get。

(7)Message (ExpC3)

在该 Get 被 Read 命令激活时,将〈ExpC3〉作为说明性消息显示在最后一行。

(8)Valid (ExpL1)

此参数用来判断按一个 Button 之后的一些动作,通常后接 UDF。

(9)When (ExpL2)

此参数用来事前判断(ExpL2)的真假值,而决定是否让 Read 去激活此按钮。

(10)Color Scheme (ExpN4)/Color (color-pair)

此参数用来让用户自行设置颜色盘或颜色。

范例

```
@1,1 Get ok Function " *H\ (Open; \\Close; \! ok; \? \ (Cancel" Size 1,8,3
```

上例中,我们看到了“\”,它表示“O”为热键,而 Close 选项不可选,“\!”表示若直接按 Ctrl + Enter 则选 OK,“\?”则是按 ESC 键所选的按键。

@...Get (单一型按钮 Radio Button)

语法

@ (行,列) GET (变量)

Function (ExpC1)/Picture (ExpC2)

[Default (Expr)]

[Size (ExpN1), (ExpN2)[, (ExpN3)]]

[Enable/Disable]

[Message (ExpC3)]

[Valid (ExpL1)]

[When (ExpL2)]



[Color Scheme <ExpN4>/Color <color-pair>]

说明

此种方法制作的按钮是非常常见的,在 FoxPro 和 Windows 中的很多地方都用到,如下例:

输出到:

- ( ) 文件
- (.) 打印机
- ( ) 预显示

这种按钮的特点是只能选择上述一种。

(1)<行,列>

定义按钮的坐标位置。

(2)Get<变量>

此变量与 Push Button 一样,若此变量为数值型,则返回值为该选项的序号,若为字符型,则返回该项的提示字符存入变量中。

(3)Function <ExpC1>/Picture <ExpC2>

① \*R:此符号表示要建立五个 Radio Button。

② N:表示选择某选项后,立即跳出 Read 命令。

③ T:表示选择某项后,立即跳出 Read 命令。其作用恰好与 N 相反。

④ H:表示要建立一个横向的 Radio Button。

⑤ V:表示要建立一个垂直的 Radio Button。

(4)Default <Expr>

若变量未事先定义时,则可以使用 Default 来自行建立该变量,且可有缺省值。

(5)Size <ExpN1>,<ExpN2>[,<ExpN3>]

这三个数值型表达式分别可用来更改 Radio Button 的高度、宽度及间隔。

(6)Enable/Disable

这两个参数则是允许或禁止 Read 命令激活。不过若加上 disable,则会使所有 Radio Button 都不能激活。

(7)Message <ExpC3>

<ExpC3>将会在该 Button 激活后显示在最后一行。

(8)Valid <ExpL1>

当用户选好某一选项时,FoxPro 会根据<ExpL1>来进行条件判断或进行其他操作,Valid 后通常接用户自定义函数(UDF)。

(9)When <ExpL2>

在 Read 要对此 Get 进行激活时会先去检查此<ExpL2>的真假值。若为真,则可激活,若为假,则不激活。When 后也通常接用户的自定义函数 UDF。

(10)Color Scheme <ExpN4>/Color <color-pair>

Color Scheme 可以让用户自行使用 1—24 颜色盘的缺省颜色,或使用 Color <colorpair>自