

第一章 FoxPro 2.6 命令和函数简介

本章完整地列出了 Microsoft FoxPro 2.6 的命令和函数,在 FoxPro 的联机帮助中也有这部分内容。当然,为节省篇幅起见,我们进行了一些压缩。建议读者多用用联机帮助,在使用 FoxPro 的过程中遇到问题时,一般总可以在联机帮助中快速找到答案。

FoxPro 2.5 和 2.6 版的特点是:具有大量新增的或者功能增强的命令、函数、预处理器伪指令及系统内存变量等。本书完整地描述了上述每一项。

1.1 FoxPro 2.5 版新增与增强的命令和函数

下表先列出了 FoxPro 2.5 版本中所有新增或功能增强的命令、函数、预处理器伪指令及系统内存变量等。每一项均标出了可用平台及增强的内容。

命令、函数、预处理器伪指令或系统内存变量	可用平台	新增	增强	增强内容
? ??	全部		✓	FoxPro for Windows 增加了 FONT 子句
@...SAY	全部		✓	在 FoxPro for Windows 中,支持 BMP 图像、OLE 对象和 RGB 色彩对,增加了下列子句:FONT、STYLE、BITMAP、CENTER、ISOMETRIC、STRETCH 和 VERB
@...GET	全部		✓	在 FoxPro for Windows 中增加了 FONT 和 STYLE 子句,支持 RGB 色彩对
@...GET--- 复选框	全部		✓	在 FoxPro for Windows 中,增加 FONT 和 STYLE 子句,支持图像、掩码和 RGB 色彩对
@...GET--- 不可见框	全部		✓	在 FoxPro for Windows 中,支持 RGB 色彩对
@...GET 列表	全部		✓	在 FoxPro for Windows 中,增加 FONT 和 STYLE 子句,支持 RGB 色彩对
@...GET--- 弹出式菜单	全部		✓	在 FoxPro for Windows 中,增加 FONT 和 STYLE 子句,支持 RGB 色彩对
@...GET--- 下推按钮	全部		✓	在 FoxPro for Windows 中,增加 FONT 和 STYLE 子句,支持图像、掩码和 RGB 色彩对
@...GET--- 单选按钮	全部		✓	在 FoxPro for Windows 中,增加 FONT 和 STYLE 子句,支持图像、掩码和 RGB 色彩对
@...GET 数码器	Windows	✓		
@...GET 正文编辑区	全部		✓	在 FoxPro for Windows 中,增加 FONT 和 STYLE 子句,支持 RGB 色彩对

命令、函数、预处理 伪指令或系统内存变量	可用平台	新增	增强	增强内容
@...FILL	全部	✓		在 FoxPro for Windows 中, 支持 RGB 色彩对
@...TO	全部	✓		在 FoxPro for Windows 中, 增加矩形、椭圆、圆、填充模式和外形大小, 支持 RGB 色彩对
AFONT()	Windows	✓		
ANSITOCHEM()	Windows	✓		
APPEND FROM	全部	✓		支持 Microsoft Excel 3.0 和 4.0 及 Paradox 3.5 和 4.0
APPEND GENERAL	Windows	✓		
BROWSE	全部	✓		在 FoxPro for Windows 中, 增加 FONT 和 STYLE 子句, 支持 RGB 色彩对
CHANGE	全部	✓		在 FoxPro for Windows 中, 增加 FONT 和 STYLE 子句, 支持 RGB 色彩对
DDEAbortTrans()	Windows	✓		
DDEAdvise()	Windows	✓		
DDEEnabled()	Windows	✓		
DDEExecute()	Windows	✓		
DDEInitiate()	Windows	✓		
DDELastError()	Windows	✓		
DDEPoke()	Windows	✓		
DDERequest()	Windows	✓		
DDESetOption()	Windows	✓		
DDESetService()	Windows	✓		
DDESetTopic()	Windows	✓		
DDETerminate()	Windows	✓		
#DEFINE...#UNDEF	全部	✓		
DEFINE BAR	全部	✓		在 FoxPro for Windows 中, 支持 RGB 色彩对
DEFINE MENU	全部	✓		在 FoxPro for Windows 中, 支持 RGB 色彩对
DEFINE PAD	全部	✓		在 FoxPro for Windows 中, 支持 RGB 色彩对
DEFINE POPUP	全部	✓		在 FoxPro for Windows 中, 支持 RGB 色彩对
DEFINE WINDOW	全部	✓		在 FoxPro for Windows 中, 增加 FONT, STYLE, AT, SIZE, FILL, FILE, ICON FILE, IN, DESKTOP, MDI, NOMDI 和 HALFHEIGHT 等子句, 并支持 RGB 色彩对
DIR/DIRECTORY	全部	✓		在 FoxPro for Windows 中, 增加了 PROMPT 子句
DISPLAY/LIST	全部	✓		在 FoxPro for Windows 中, 增加了 PROMPT 子句

命令、函数、预处理 伪指令或系统内存变量	可用平台	新增	增强	增强内容
_DOS	全部	✓		
FONTMETRIC()	Windows	✓		
GETDIR()	全部	✓		
GETFONT()	Windows	✓		
#IF...#ENDIF	全部	✓		
IMPORT	全部		✓	支持 Microsoft Excel 3.0 和 4.0 及 Paradox 3.5 和 4.0
LABEL	全部		✓	在 FoxPro for Windows 中, 增加了 PROMPT 子句
MAC	全部	✓		
MODIFY COMMAND/FILE	全部		✓	在 FoxPro for Windows 中, 增加了 SAME 子句
MODIFY GENERAL	Windows	✓		
MODIFY MENU	全部		✓	在 FoxPro for Windows 中, 增加了 SAME 子句
MODIFY WINDOW	Windows	✓		
MOVE WINDOW	全部		✓	增加了 CENTER 子句
OBJECTANSI()	Windows	✓		
READ()	全部		✓	在 FoxPro for Windows 中, 支持 RGB 色彩对
REPORT	全部		✓	在 FoxPro for Windows 中, 增加了 PROMPT 子句
SELECT-SQL	全部		✓	在 FoxPro for Windows 中, 增加了 NOWAIT, PREFERENCE, PROMPT 和 TO SCREEN 子句, 查询结果可送往一个 Browse 窗口
SET CLOCK	全部		✓	在 FoxPro for Windows 中, 增加了 STATUS 子句
SET DEVELOPMENT	全部		✓	在 FoxPro for Windows 中, 在执行 READ 命令期间可中断程序运行
SET DEVICE	全部		✓	在 FoxPro for Windows 中, 增加了 PROMPT 子句
SET HELP	全部		✓	在 FoxPro for Windows 中, 增加了 Windows 型帮助
SET KEYCOMP	Windows	✓		
SET PALETTE	Windows	✓		
SET PRINT	全部		✓	在 FoxPro for Windows 中, 增加了 PROMPT 子句
SET STATUS	全部		✓	在 FoxPro for Windows 中, 增加了 TIMEOUT 子句
SET STATUS BAR	Windows	✓		
SHOW GET	全部		✓	在 FoxPro for Windows 中, 支持 RGB 色彩对
SHOW GETS	全部		✓	在 FoxPro for Windows 中, 支持 RGB 色彩对
SHOW OBJECT	全部		✓	在 FoxPro for Windows 中, 支持 RGB 色彩对

命令、函数、预处理 伪指令或系统内存变量	可用平台	新增	增强	增强内容
SHOW WINDOW	全部		✓	REFRESH 子句更新备注编辑窗口
SYS(20)	全部	✓		
SYSMETRIC()	Windows	✓		
_TRANSPORT	全部	✓		
TYPE	全部		✓	在 FoxPro for Windows 中,增加了 PROMPT 子句
TEXTWIDTH()	Windows	✓		
_UNIX	全部	✓		
USE	全部		✓	增加了 SHARE 子句,并支持 0 作为工作区号
WFRONT()	Windows	✓		
_WINDOWS	全部	✓		

1.2 FoxPro 2.6 版新增与增强的命令和函数

为了使 FoxPro 2.6 与 dBASE IV 相兼容, FoxPro 2.6 新增了三十多个函数和一些命令,而且还修改了另外一些命令的功能。下面列出了 FoxPro 2.6 所新增的函数:

BARCOUNT()
 BARPROMPT()
 DESCENDING()
 FDATE()
 FLDXCOUNT()
 FLDLIST()
 FOR()
 FTIME()
 HOME()
 ID()
 ISBLANK()
 ISREADONLY()
 KEYMATCH()
 MEMORY()
 NORMALIZE()
 OBJVAR()
 PADPROMPT()
 PCOUNT()
 PRINFO()
 RUN()

```

SET("ATTRIBUTES")
SET("DIRECTORY")
SET("DISPLAY")
SET("LIBRARY")
SET("MBLOCK")
SYS(1037)()
TAG()
TAGCOUNT()
TAGNO()
UNIQUE()
WINDOW()

```

下面是 FoxPro 2.6 新增的命令：

```

ASSIST
BLANK
ON BAR
ON EXIT BAR
ON EXIT MENU
ON EXIT PAD
ON EXIT POPUP
ON MENU
ON MOUSE
ON PAD
ON POPUP
SET DIRECTORY
SET FIELDS
SET FORMAT
SET KEYCOMP
SET MBLOCK TO
SET VIEW

```

下面是 FoxPro 2.6 修改了功能的一些命令：

```

* APPEND FROM
* APPEND FROM ARRAY
BROWSE
* COPY STRUCTURE
* COPY STRUCTURE EXTENDED
* COPY TO
* COPY TO ARRAY

```

```
*CREATE REPORT
*CREATE SCREEN
*DISPLAY
*EXPORT
*GATHER
*JOIN
KEYBOARD
*LIST
ON SELECTION POPUP
*SCATTER
SET COMPATIBILITY
SET FIELDS
*SORT
SQL-SELECT
*TOTAL
```

在 FoxPro 2.6 中,凡是支持 Field 子句的命令(以“*”号作为标记者)都可带有 Like 子句或 EXCEPT 子句。此外,FoxPro 2.6 还增加了 .ASSIST 和 .SHELL 这两个系统内存变量。

1.3 新增的数据排序函数及命令

很多时候,你需要检索某一索引属性。在 FoxPro 中始终能够使用 KEY() 函数和 TAG() 函数来检索索引码及标识名。但是,FoxPro 2.6 以前的版本却不能确定一个索引是以升序还是以降序排列,以及是否唯一索引。幸运的是,为了实现与 dBASE IV 的兼容,Microsoft 公司为 FoxPro 2.6 增加了以下 7 个用于检索索引属性的参数。

函数	类型
DESCENDING([<expFN>],[<expN>],[<expWA>]])	L
FLDCOUNT([<expWA>])	N
FOR([<expN>],[<expWA>]])	C
KEYMATCH(<exp>,[<expN1>],[<expWA>]])	L
TAGCOUNT([<expFN>],[<expWA>]])	N
TAGNO([<expC>],[<expFN>],[<expWA>]])	N
UNIQUE([<expFN>],[<expN>],[<expWA>]])	L

任选的函数参数(<expFN>)用于给出索引文件名称,它通常用于索引文件当前并未使用的情形。

任选参数(<expWA>)可用于定义一个包含有一个数据库文件和若干活动索引文件的工作区。如果不指定工作区,则缺省选定当前工作区。

参数(<expN>)用于表示 IDX 文件的序号或其在索引目录中的索引标识。

KEYMATCH() 函数中的任选函数(<expN1>)也用于表示 IDX 文件的序号或其在索引目录

中的索引标识。

TAGNOO函数中的(expC)参数用于表示索引标识的名称或者所要求索引位置处的IDX文件。对于所有这些函数而言,顺序的建立是依据USE命令或SET INDEX TO命令所指定的索引顺序。通常,IDX文件的序号在前,接着是结构化的CDX文件,再后是任意指定的CDX文件。

如果函数参数中所给定的索引序号是无效的,则函数将返回空串。如果参数中给定一个无效的别名,则屏幕上将显示出错信息“Alias not found”。

1.4 菜单系统命令和函数

为了实现与dBASE IV V2.0的兼容,FoxPro 2.6新增了一些命令和函数,以支持用户定义的菜单系统。语言的增强给FoxPro的开发者提供了更大的兴趣和机会。

1.5 其他函数

FoxPro 2.6为了确保与dBASE IV相兼容,增加了一些新的函数和命令。

例如,数据包括已知的和未知的值。在很多应用中,处理时要用到的数据库记录并不总是有值的。为了解决这一问题,也为了实现兼容性,FoxPro 2.6借用了dBASE IV的ISBLANK()函数。这一函数可以更好地支持空字段或null字段的概念。

再如,一些开发者从不使用PACK命令来物理地删除数据库文件记录,他们简单地重用被删除的记录。为了重新利用记录,FoxPro从dBASE IV中借用了BLANK命令以提供恢复能力。

1.6 dBASE IV 2.0和FoxPro 2.6的语言差别

FoxPro 2.6 for Windows与dBASE IV基本兼容,但也存在着一些区别。此外,FoxPro 2.6不支持dBASE IV的事务处理命令和某些网络命令。

dBASE IV 2.0与FoxPro 2.6 for Windows的语言差别说明如下:

命令和函数	二者区别
ACatalog ManagerESS()	FoxPro不支持该函数,但不出错。
@... GET	FoxPro忽略REQUIRED关键字。
APPEND FROM	FoxPro忽略REINDEX关键字。
APPEND FROM ARRAY	FoxPro忽略REINDEX关键字。
BARCOUNT()	在dBASE IV中,BARCOUNT()计算并返回所定义的弹出式菜单条的数目,而无论该弹出式菜单是否被激活;在FoxPro中,BARCOUNT()计算并返回活动的弹出式菜单的菜单条的数目。如果一个弹出式菜单已经定义但未被激活,那么BARCOUNT()返回0。
BEGIN TRANSACTION	FoxPro不支持该命令并显示“不可识别命令”错误信息

BROWSE	信息。
CHANGE()	FoxPro 忽略 NOORGANIZE 关键字。
COMPLETED()	在 FoxPro 中,该函数永远返回逻辑“假”值(.F.)。
CONVERT()	在 FoxPro 中,该函数永远返回逻辑“假”值(.F.)。
	FoxPro 不支持该函数并显示“不可识别命令”错误信息。
END TRANSACTION	FoxPro 不支持该函数并显示“不可识别命令”错误信息。
FCREATE()和 FOPEN()	在 dBASE IV 中,这两个函数的第 2 个参数可以是“R”(Read-Only)、“W”(Write-only)、“A”(Append-only)、“RW”或“WR”(Read 和 Write)、“RA”或“AR”(Read 和 Append)等几种形式。
	FoxPro 支持上述文件属性。
FCREATE()	参数为“RW”或“WR”时,FoxPro 创建一个可读可写的文件并设置 MS-DOS 档案属性;参数为“A”时,FoxPro 创建一个可读可写的文件并设置 MS-DOS 档案属性;参数为“RA”或“AR”时,FoxPro 创建一个可读可写的文件并设置 MS-DOS 档案属性;参数为“R”时,FoxPro 创建一个只读文件并设置 MS-DOS 的只读属性;
FOPEN()	参数为“RW”或“WR”时,FoxPro 以可读可写方式打开一个文件并将文件指针定位于文件头;参数为“A”时,FoxPro 以只写方式打开一个文件并将文件指针定位于文件尾部;参数为“RA”或“AR”时,FoxPro 以可读可写方式打开一个文件并将文件指针定位于文件尾部;参数为“R”时,以只读方式打开一个文件并将文件指针定位于文件头。
FLDLIST()	在 dBASE IV 中,该函数返回一个以数据库文件别名和“-”操作符为前缀的字段名;在 FoxPro 中,该函数返回一个以数据库文件别名和圆点(.)操作符为前缀的字段名。
ID()	在 dBASE 中,ID()返回当前正运行 dBASE 的机器的名称;在 FoxPro 中,ID()返回当前正运行 FoxPro 的机器的名称及其在网络中的机器号。如果 FoxPro 不是处于网络环境下,则 ID()返回机器名和数字 1。
NETWORK()	该函数在 FoxPro 中的返回值始终为 .T.,即 FoxPro 并不支持该函数。
PROTECT	FoxPro 不支持该命令并生成“不可识别命令”错误信息。

RESET	在 FoxPro 中, RESET 命令被忽略。
ROLLBACK	FoxPro 不支持该命令并生成“不可识别命令”的出错信息。
RUN()	FoxPro 不支持 dBASE IV 的 RUN() 函数, 但 FoxPro 提供了 RUN 命令。
SET('BORDER')	在 FoxPro 中, 该函数返回边界字符, 而在 dBASE IV 中, 该函数返回边界字符的 ASCII 码。
SET('CATALOG')	FoxPro 始终返回“OFF”状态。
SET DETRAP	FoxPro 忽略此命令。
SET ENCRYPTION	SET ENCRYPTION ON/OFF 命令在 FoxPro 中被忽略。dBASE IV 有很好的加密技术, 但这一功能在 FoxPro 中不被支持。
SET("INSTRUCT")	FoxPro 始终返回“OFF”状态。
SET FIELDS	在 dBASE 的 SET FIELDS 命令中, 可以为其中的任一字段设置“/R”开关, 而这一开关在 FoxPro 2.6 中被忽略。
SET('FIELDS', 1)	当 FIELDS 被设置为 LOCAL 并且 COMPATIBLE 被设置为 FOXPLUS 或 OFF 时, SET('FIELDS', 1) 函数所返回的字段名称不带有数据库文件别名。
SET IBLOCK	FoxPro 忽略该命令。
SET("INSTRUCT")	FoxPro 始终返回“OFF”状态。
SET("LDXCHECK")	FoxPro 始终返回“OFF”状态。
SET LDXCHECK	FoxPro 忽略该命令。
SET MESSAGE	FoxPro 忽略该命令中的 AT 子句。
SET MULTLOCKS ON	多记录锁选项在 dBASE IV 中是隐含的, 但在 FoxPro 中却需要显式地建立。
SET("PRECISION")	FoxPro 始终返回数值 16。
SET("SQL")	FoxPro 始终返回状态“ON”。
SET ("TRAP")	FoxPro 始终返回状态“ON”。
SET("VIEW")	FoxPro 始终返回“(空串)”。
USE	FoxPro 忽略 NOSAVE 和 NOLOG 关键字。

FoxPro 2.6 与 dBASE IV 除了存在着语言和语法上的差别外, 二者在功能实现方式上也存在一些重大的差别。例如, 在 dBASE IV 中, 向过程和函数传递参数的方式都是通过引用传递, 而在 FoxPro 中, 过程的参数传递是通过引用, 函数的参数传递则是通过值。如果希望以引用方式传递参数, 只需在程序中增加一条 SET UDF PARMs TO REFERENCE 语句。

当 dBASE IV 从一个 UDF 中返回时, 通常会重新选取在调用该 UDF 之前处于活动状态的那个工作区。而在 FoxPro 中, 如果调用了—个 UDF 并且该 UDF 在执行时重新选取了工作区, 那么当控制从该 UDF 返回到调用程序之后, 系统并不会去自动恢复调用前的工作区。

dBASE IV 和 FoxPro 在交互环境下各自使用的键盘快捷键也不相同。这样, 如果一个

dBASE IV 应用程序使用了 READKEY 或 LASTKEY 等读键命令,那么它运行于 FoxPro 中就会产生问题。如果不满意 FoxPro 的键盘设置,那么可以尝试修改 FOXPLUS.KEY 宏文件以使键盘设置与 dBASE IV 相统一,然后只需在应用程序中增加一条 RESTORE MACROS FROM FOXPLUS.FKY 语句即可。

警告:如果试图将标准块大小(1BLOCK 和 MBLOCK)的 dBASE IV 索引文件或备注文件转换为 FoxPro 格式,这一转换操作将不可能成功。对于索引文件,你需要删除 MDX 文件并重新创建 FoxPro 的索引标识。另一种方案是使用 dBASE IV 以缺省的 1BLOCK 设置来重新创建 MDX 文件,然后即可在 FoxPro 中将 MDX 文件转换为 FoxPro 格式。

1.7 dBASE IV 模板编程语言

dBASE IV 的模板语言完全不同于 dBASE IV 语言或其他的任何 Xbase 对话,它使用了完全不同的语法。dBASE 模板语言无法使用于 FoxPro 环境,因此如果需要,唯有重写程序代码。不过,这些模板的功能仅仅是为 MS-DOS 创建基本的 Xbase 表格和报表程序。FoxPro 2.6 for Windows 可以将 dBASE IV 的表格和报表设计文件直接转换为 FoxPro 可识别的格式。由于 Windows 与 MS-DOS 所存在的差别,你可能需要修改屏幕,而且还可能需要修改报表以利用各种字段。无论要进行何种改变,通过 FoxPro 的强大工具都是不难实现的。

熟悉 FoxPro 的屏幕代码模板是比较容易的,因为你不需要学习另外的模板语言。FoxPro 语言支持编程。可用于编程的 FoxPro 命令包括:

```
\\ \\
SET TEXTMERGE DELIMITERS
SET TEXTMERGE
TEXT
_PRETEXT
TEXT ... ENDTEXT
```

1.8 文件扩展名

如果你要转换为 FoxPro 2.6 for Windows 的 dBASE IV 应用程序中包含有硬代码文件扩展名,你可能需要改变其中的一些扩展名。下面列出了各种类型的 dBASE IV 文件扩展名以及与之相对应的 FoxPro 2.6 文件扩展名。通过 Run 菜单下的 Convert dBASE Files... 选项可以完成扩展名的转换:

dBASE 文件	FoxPro 文件	文件类型
.DBF	.DBF	数据库文件(无变化)
.DBF	.FPT	备注文件
.FMT	.FMT	.PRX 格式文件
.FRG	.FRG	报表程序文件(无变化)
.FRM	.SPR	屏幕程序文件

.LBG	.LPG	标签程序文件(无变化)
.LBL	.LBX	.LBT 标签设计文件(无变化)
.MDX	.CDX	多索引文件
.NDX	.IDX	索引文件
.PRG	.PRG	程序文件(无变化)
.QBE	.QPR	查询程序文件

第二章 FoxPro 命令和函数详解

5
1

(1)%

作用:求模操作符——返回一个数字型表达式除以另一个数字型表达式所得的余数。

语法:⟨expN1⟩%⟨expN2⟩

参数:⟨expN1⟩被除数

⟨expN2⟩除数

相关函数:MOD()

说明:求模操作符% 返回一个数字型表达式除以另一个数字型表达式所得的余数。求模操作符和 MOD() 返回相同的结果。

求模操作符% 是一个和+(加)、-(减)、*(乘)、/(除)、^ (取幂)一样的算术运算符。当这些操作符在一个数字型表达式中混合出现时,% 具有与*和/相同的优先级。

参数:

⟨expN1⟩

⟨expN1⟩是被除数(被除的数字型表达式),⟨expN1⟩中的小数位数决定了结果中的小数位数。

⟨expN2⟩

⟨expN2⟩是除数(用来除被除数⟨expN1⟩的数字型表达式)。如果⟨expN2⟩的值为正,则结果为正;如果⟨expN2⟩的值为负,则结果为负。⟨expN2⟩不能为零。

例子:

? 36%10

6

? (4*9)%(90/9)

6

? 25.250%5.0

0.250

? IIF(YEAR(DATE()))%4=0,'Summer Olympics this year';

,'No Summer Olympics this year')

(2)&

作用:宏替换。

语法:&⟨memvar⟩[.⟨expC⟩]

说明:宏替换把一个内存变量或一个数组元素的内容当作一个字符串文字量来处理。当一个字符型的内存变量或数组元素前出现一个&时,该变量或元素的内容就替换这个宏引用。宏替换可以在任何能接受字符串文字量的命令或函数中使用。

可能的话,用户最好使用名表达式而不用宏替换。因为名表达式的作用与宏替换类似,只不过名表达式只能将字符串作为名来传递。如果一个命令或函数接受名(文件名、窗口名、弹出式菜单名…),那么就可以使用一个名表达式来极大地提高处理速度。有关名表达式,请参阅本书“FoxPro 语言概述”一章。

参数:

⟨memvar⟩

这是在宏替换中被引用的内存变量名或数组元素名,不含用来将内存变量与字段区别开来的 M_ 前缀(这将导致语法错)。宏不能超过 FoxPro 中允许的最大语句长度。

出现在 DO WHILE, FOR 和 SCAN 中的宏替换语句仅在循环的一开始进行一次计算求值,而在随后的迭代中不再重新计算求值。因而在循环过程中,即使内存变量或数组元素发生了变化,这些变化也都不会对该循环有任何影响。

· (expC)

任选的, (句点) 定义符和字符型表达式 (expC) 是用来给一个宏添加额外字符的。用 · (expC) 添加到宏上的字符型表达式 (expC) 本身也可以是一个宏。

(3) \$

作用: 如果一个字符型表达式包含在另一个字符型表达式中,则返回真值。

语法: (expC1) \$ (expC2)

参数: (expC1) 搜索字符

(expC2) 被搜索字符

返回值: 逻辑值

相关函数: AT()

说明:

如果一个字符型表达式包含在另一个字符型表达式中,则该函数返回真值 (.T.); 否则返回假值 (.F.)。该函数是不可 Rushmore 优化的。

参数:

(expC1), (expC2)

在字符型表达式 (expC2) 中搜索 (expC1) 的出现。如果在 (expC2) 中找到 (expC1), 则返回 .T.; 如果在 (expC2) 中没有找到 (expC1), 则返回 .F.。该函数区别大小写字母。

(expC1) 和 (expC2) 可以是字符类型的内存变量或数组元素、字符类型的字段、字符串文字量或任意长度的备注字段。

对于备注字段可以像对字符型表达式、表/.DBF 中的字段、内存变量或数组元素一样进行操作。

例子: 下面的例子产生一个名叫 MEMOTEST 的表/.DBF, 它包含一个备注字段。有三个记录要添加到表/.DBF 中, 并打开一个 Browse(浏览)窗口来显示这 3 个记录。\$ 用来显示那些包含字符串“FOX”的记录。随后, 为本例创建的那些文件将被删除掉。

```
CLOSE DATABASES
```

```
CLEAR
```

```
CREATE CURSOR memotest(text C(3), memo M)
```

```
INSERT INTO memotest (text,memo) VALUES('Fox','Fox')
INSERT INTO memotest (text,memo) VALUES('Cat','Cat')
INSERT INTO memotest (text,memo) VALUES('FOX','FOX')
```

```
LIST FIELDS memo,text FOR 'FOX' $ UPPER(memo)
USE
```

2

(4) =

作用:计算一个或多个表达式的值。

语法:=(expr1)[,(expr2)...]

相关函数:?,用户定义函数

说明:=命令计算一个或多个表达式(expr1),(expr2)...并舍弃返回值。有时我们需要一个FoxPro函数或用户定义函数有我们所期望的功能,并不需要将该函数的返回值赋给一个内存变量、数组元素或字段,这时=命令就显得特别有用。

“=”号还有两个不相关联的用途:在逻辑表达式中它被用作一种操作符来进行比较,或给内存变量和数组元素赋值。在这两种情况下,“=”是一种操作符而不是一个命令。

(5) \ \

作用:输出文本行。

语法:\(text line) | \ \ (text line)

相关函数:PRETEXT,SET TEXTMERGE,SET TEXTMERGE DELIMITERS,TEXT,TEXT...ENDTEXT

说明:\和\ \命令方便了FoxPro中的正文合并。正文合并可以使用户将正文输出到一个文件中,从而生成信件或程序。

使用\和\ \将一正文行输出到当前的正文合并输出文件和屏幕上。SET TEXTMERGE用来指定正文合并输出文件。如果正文合并输出不是面向某一文件,则正文行仅输出到桌面(对于FoxPro for MS-DOS而言)或FoxPro主窗口(对于FoxPro for Windows而言)。SET TEXTMERGE NO SHOW用于取消到桌面或FoxPro主窗口的输出。

使用\,输出行以一个回车换行符打头。使用\ \,输出行不以回车换行符打头。

\和\ \前面的空格不包含在输出行中,但\和\ \后面的空格则包含在输出行中。

用户可在正文行中嵌入一个表达式。如果表达式放在正文合并界限符(缺省为((和)))中,并且TEXTMERGE为SET ON,则表达式被计算并且它的值将作为正文输出。

例子:

```
USE customer
SET TEXTMERGE ON
SET TEXTMERGE TO letter.txt
\((CLOW(DATE()))),(CMONTH(DATE()))
\ \ ((DAY(DATE()))),(YEAR(DATE()))
\
```

```

\
\Dear<contact>
\
\Additional Text
\
\Thank you,
\
\XYZ Corporation
CLOSE ALL
MODIFY FILE letter.txt NOEDIT

```

(6)? |??

作用:对表达式求值并输出结果。

语法:? |?? <expr1>

```

[PICTURE <expC1>]||[FUNCTION <expC2>]
[AT <expN1>]
[FONT <expC3>[, <expN2>]
[STYLE <expC4>| <expR2>]
[, <expR3>], ...

```

注意事项:FONT子句仅适用于 FoxPro for Windows。

相关函数:???, @...SAY, SET MEMOWIDTH, SET PRINTER

说明:?和??对表达式求值并将结果输出到桌面(对于 FoxPro for MS-DOS 而言)、FoxPro 主窗口(对于 FoxPro for Windows 而言)、活动的用户定义窗口或打印机。

如果 SET PRINTER 为 ON, 则表达式的结果输出将送向打印机和桌面、FoxPro 主窗口或活动的用户定义窗口。如果 SET PRINTER 为 ON 而且 SET CONSOLE 为 OFF, 则结果输出仅送向打印机。

子句:

?

单独一个问号在输出表达式的结果前先发送一个回车换行符。除非另外指定一个功能码(<expC2>)或 ALIGNMENT 系统内存变量, 否则结果将被显示在桌面、FoxPro 主窗口或活动的用户定义窗口的下一行, 并在一页的左边界处打印。

如果省去表达式, 将打印一个空行。而当有多个表达式时, 各表达式的结果之间将置一空格。

??

两个问号将表达式的结果输出到桌面、FoxPro 主窗口或活动的用户自定义窗口或打印机的当前行当前位置, 但在输出表达式的结果前并不发送一个回车换行符。

PICTURE <expC1>

如果选用了 PICTURE 子句, 则 <expR1> 的结果将按 <expC1> 指定的格式进行显示。(<expC1> 可由功能码、图像码或二者的组合所构成。

凡在@...SAY中能用的图像码和功能码,这里也同样适用。参见@...SAY中所列的所有图像码和功能码。

功能码影响表达式结果的整个格式,而图像码只对结果中的单个字符起作用。如果在(expC1)中用到功能码,则它们必须出现在图像码之前,并且以@打头。多个功能码可以立即跟在@之后,不必嵌入空格。最后一个功能码之后必须有一个以上的空格,这些空格标志着功能码的结束和图像码的开始。

FUNCTION <expC2>

FUNCTION 字符型表达式(expC2)提供了另外一种在?和??的输出中包含功能码的方法。如果选用了FUNCTION子句,则功能码不以@打头。仅在PICTURE中用到的功能码必须以@打头。

V(n)

V(n)是一个能在?和??中使用的特殊的功能代码。该代码允许字符型表达式的输出结果在有限内垂直展览。(n)为输出中的列数。

```
? 'This is an example of how the V(n) function works.';
```

```
FUNCTION 'V10'
```

AT(expN1)

AT用于指定输出显示中的列数(expN1)。该选项用来将输出列对齐,从而生成一个表/DBF。数字型表达式(expN1)可以是一个返回数字值的UDF。

FONT(expC3)[,<expN2>]——只适用于FoxPro for Windows

字符型表达式(expC3)是字模的名字,数字型表达式(expN2)是字模的大小。

如果选用了FONT子句但是省去了字模的大小(expN2),则使用10点的字模。

如果省去了FONT子句而将?|??的输出置于FoxPro主窗口,则在输出中使用FoxPro主窗口的字模。如果省去了FONT子句而将?|??的输出置于一个用户定义窗口,则在输出中使用该用户定义窗口的字模。

如果用户指定的字模不存在,Windows将用一个类似的字模来代替。

STYLE <expC4>|<expC2>

在FoxPro for Windows中,选用STYLE子句来指定?|??输出的字体。一个字模可以使用哪些字体由Windows决定。如果用户指定的字体不存在,Windows将用一个类似的字体来代替。

字体由<expC4>指定。如果省去了STYLE子句,则使用标准字体。

expC4	字体
B	黑体
I	斜体
N	普通
O	轮廓线
Q	不透明
S	阴影
-	删除线
T	透明

U 下划线

可以使用一个以上的字符来指定字体的组合。

在 FoxPro for MS-DOS 中,STYLE 子句用于指定由? 和?? 发送至打印机上的输出的字体和字模的大小。在现行的 FoxPro 打印机驱动程序中,STYLE 子句中的表达式(expr2)被传递给 PDI)对ST,PDI)OBJECT 和 PDI)OBJEND 这三个过程。这些过程决定了? 和?? 输出的打印方式。当? 和?? 的输出被送至桌面或用户定义窗口时,STYLE 子句将被忽略。

例子:

```
? 15 * (10+10)
300

? 'Welcome to ' PICTURE '@:!'
?? 'FoxPro'
WELCOME TO FoxPro
```

(7)???

作用:将输出直接送往打印机。

语法:??? (expC)

相关函数: ? |??, @, ...SAY, @, ...GET, CHR()

说明:三个问号将(expC)的内容直接送往打印机,而不增加打印的行和列。??? 允许用户将打印机控制码直接送给打印机而不推进打印头。

打印机控制码允许用户重置打印机,改变字样和字模大小,设置或取消粗体打印等。这些控制码可由用户所使用的打印机的可打印字符和不可打印的字符组成。用户可以用下面几种不同的方式将控制码直接送往打印机。

- 使用 CHR() 将 ASCII 字符直接送给打印机。
- 使用花括号{ }发送打印机控制码或 ASCII 字符。
- 使用 CHR() 和花括号的组合。
- 控制码既可在打印开始之前,也可在打印结束之后,由系统内存变量 _PSCODE 和 _PECODE 传送给打印机。详见本书“系统内存变量”一章。

不同打印机的打印机控制码也是不同的。有关打印机控制码请用户参考与打印机配套的手册。

(8)@, ...SAY

作用:在指定的行列处显示输出。

语法:@, (row, column)

```
SAY (expr)
[FUNCTION (expC1)]
[PICTURE (expC2)]
[SIZE (expN1), (expN2)]
[FONT (expC3)[, (expN3)]]
```