命令模式

MACro name ["title"] [/FUNction]

Defines a macro command.

定义一宏命令。

MACro name EDIT

Allows editing an existing macro.

编辑一存在的宏命令。

MACro [/List:pattern]

Lists the names of all matching macro commands (screen only).

屏幕显示匹配的宏命令名称。

参数说明

name

The macro command name; (no embedded spaces or other separators).

宏命令名称；（不得包含空格和分隔符）。

"title"

A descriptive title displayed in place of the macro name on template and LEw button labels.

宏命令描述性的说明，它会作为标签显示在模板或者装载编辑器的按键上。

/FUNCTION

Identifies the macro as a function whose initially-empty value can be SET by the macro then retrieved using name([parameter],...) calls within braced expressions and variable assignments. Useful SET operator keywords are predefined as functions so cannot be redefined.

将宏标识为一个函数，其初始空值可由宏 SET 使用大括号表达式和变量赋值中的 name（[参数],...） 调用进行检索。有用的 SET 运算符关键字预定义为函数，因此无法重新定义。

/LIST: pattern

Restricts list to only names matching pattern, which may include wildcard (* and ?) characters.

显示和 pattern 相匹配的宏命令名称，可以包含通配符* 和 ？。

Definition: A macro command (or "macro" for short) is a user-defined command made up of a series of command input lines.

定义：一条宏命令（或简称“宏”）是用户定义的由多条命令行组成的命令。

See the EXECUTE command for a discussion of how a macro is executed as a command. See also the WRITE command for saving macros to a file, the LIST command for listing macro contents, and the CLEAR command for deleting all macro definitions.

可以参看命令 EXECUTE 了解宏命令是如何被运行的。可以参看命令 WRITE 了解宏命令如何被保存到文件。可以参看命令 LIST 了解如何显示宏命令内容。可以参看命令 CLEAR 了解如何删除所有定义的宏命令。

Special macros are described in the following sections: AUTOEXEC (Main Program), ESC (ERROR command), HALT and HELP (MENU command), and IDLE and WAITING (WAIT command).

一些特殊的宏命令可在如下章节中查看：

AUTOEXEC(主程序)，ESC（命令 ERROR），HALT 和 HELP（命令 MENU），IDLE 和 WAITING(命令 WAIT)。

Operation

操作

MACRO definition establishes a new macro command name and begins storing subsequent lines of command input as the body of the macro, replacing any old definition. All subsequent input is taken as part of the macro definition until a line is encountered which begins with a single slash (/). Hence a MACRO definition has the following form:

MACRO 命令将把它定义为新的宏命令，并开始存储下面的输入做为命令的主体内容，同时替代任何老的命令定义。后面所有的输入都作为宏命令的内容，直到遇到只有单独一个斜杠“/”的行。因此宏命令定义的格式如下：

```
MACRO name                  `MACRO 名称
  first line of commands    `第一行命令
  ...
  last line of commands     `最后一行命令
/                           `斜杠
```

As the body of a macro is being assembled, all input lines are stored without interpretation until a line beginning with a slash is encountered. First, the slash is removed. Then, if an additional slash is present at the beginning of the line, it is stored as part of the macro body; otherwise the macro definition is complete. This allows other macro definitions to occur within a macro. For example,

作为宏命令的主体内容，所有的输入都被存储直到遇到“/”行。首先，此斜线将被除掉。如果，如果出现另外一以斜线“/”开头的行，它将被储存为宏命令中主体内容的一部分。否则宏命令主体内容结束。这样可允许在一个宏命令中嵌套定义另一宏命令。例如：

```
MACRO XXX
   MACRO YYY
      MESSAGE "This is macro YYY speaking.
   //
/
```

Macro XXX, when executed, spawns another macro named YYY. This nesting may occur to as many levels as desired.

当运行宏命令 XXX 时，会读入宏命令 YYY 的定义. 这种形式可以嵌套无数多个子宏命令。

If parameters are involved in a macro definition, they are represented in the macro body by 2- character strings of the form "%n", where n is a digit from 1 to 9. For example,

如果宏命令定义中引用了参数，参数将以%n 的格式出现在宏命令中，其中 n 为 1 到 9 的数字，例如：

```
    MACRO MSDRAFT
       DRAFT = %1 @ %2
     /
```

defines a macro named MSDRAFT which replaces %1 and %2 with the first two parameters. If ".MSDRAFT 12 MS" is entered, then "DRAFT = 12 @ MS" is actually executed.

此处定义一个宏名为 MSDRAFT，调用宏时可用两参数，来替代%1 和%2 所在的位置。如果运行语句“.MSDRAFT 12 MS”，那么实际上等效于运行 DRAFT = 12 @ MS。

Parameters can appear in any order within the body of the macro definition. The order that the actual parameter values are given on the EXECUTE command line determines their identification with %1, %2, etc. for replacement within the macro body. If fewer than n actual parameters are specified at execution time, then any use of %n is removed.

参数在宏命令的定义中可以任何的顺序出现。参数在 EXECUTE 命令中的实际顺序决定了在宏命令中被引用的顺序（%1,%2）。如果在执行命令中实际的参数个数小于宏命令中需要的参数%n，那么将删除任何的参数%n。

When a macro is defined within an outer macro (as in the example with YYY above), care must be taken to avoid unintended parameter substitution within the inner macro when the outer macro is executed. To define %1 within the inner macro, %%91 should be put in place of %1 (assuming the outer macro will always have less than 9 actual parameters). When the outer macro is executed, the %9 is replaced by nothing, leaving %1 in the inner macro.

当一个内部的宏定义在另一个外部的宏命令中时（正如在上面的例子：定义宏 YYY），当外部的宏运行时，一定要注意无意的错误参数替代。定义内部的宏参数%1 时，应用%%91 来代替%1（假设外部宏的实际参数总是少于 9 个）。当运行外部宏时，%9 将被忽略，剩下%1 在内部宏中。

Macro execution is terminated when the end of the macro is reached or an EXIT command is encountered. If a macro named XXX executes the command EXIT XXX, then it loops to the top and continues execution from the first line in XXX.

宏命令运行到命令的结尾或运行到 EXIT 命令时便会结束，如果宏 XXX 中运行了命令 EXIT XXX，那么执行的命令会回到宏的顶部重新从宏 XXX 第一行运行。

An empty macro definition is not retained; i.e.:

不会保留任何空的宏命令：

```
    MACRO ZZZ
     /
```

does not retain ZZZ as a macro command. In fact, if ZZZ had already existed, it would be effectively eliminated. This is the technique used to delete unwanted macros.

ZZZ 不会被保存为宏命令。实际上，如果宏 ZZZ 已经存在，它将会把宏 ZZZ 删掉。这可以用来删掉不需要保存的宏命令。

A macro defined with the /FUNCTION parameter can store its function value using a SET command with its name instead of a variable before the "=" sign.

使用 /FUNCTION 参数定义的宏可以使用带有其名称的 SET 命令而不是“=”符号前的变量来存储其函数值。

Functions can be executed and their values used within braced expressions by following their names with a parenthesized comma-separated parameter list or "()" if none. So {function(1,2,3)} executes .function 1,2,3 before returning the function value. This value can also be directly assigned to a variable X using X:=function(1,2,3) without requiring braces. See the VARIABLE and SET commands for more details.

可以在带括号的表达式中执行函数及其值，方法是在名称后面加上括号括起来的逗号分隔的参数列表，如果没有，则使用“（）”。所以 {function（1，2，3）} 在返回函数值之前执行 .function 1，2，3。这个值也可以使用 X：=function（1，2，3） 直接分配给变量 X，而无需大括号。有关更多详细信息，请参阅 VARIABLE 和 SET 命令。

MACRO name EDIT brings an existing macro into the editor (or begins a new macro). After the editing session is complete the new or revised macro definition replaces the old one. The initial "MACRO name" line and final "/" line around the contents of the named macro appear during editing. The "MACRO name" line must still be present when done editing, but the name can be changed in order to modify the contents of a macro and save it with a new name.

MACRO name EDIT 可以把已存在的宏调入到编辑器中进行编辑（或定义一个新的宏）。编辑后的宏会取代原来的宏。在编辑时"MACRO name"行和"/"行以及宏内容行都会显示。在编辑结束时，必须有 MACRO name 行。但可以改变宏的名称，以用来修改一个宏的内容并将其另存为一个新的宏。

Display Output:

显示输出

All existing macro names (on the screen only) if no parameters are present.

如果无附加参数，会屏幕显示所有当前存在的宏命令名称。

Nondisplay Output:

无显示输出

none.

无

Examples

样例

Defining a macro command called CASE performs calculations with a given deck load:

定义宏 CASE，用来基于给定的甲板载荷进行计算：

```
MACRO CASE
PAGE
\ \ \**** Deck load is %1 TONS ****\
ADD "Deck Load" %1, 0, 0, 25
SOLVE
STATUS
RA
/
```

Using macro CASE:

运行宏 CASE：

```
.CASE 1000 | .CASE 2000 | .CASE 3000
```

Editing macro CASE:

编辑宏 CASE：

```
MACRO CASE EDIT
```

Defining a macro named TESTLOOP which performs TEST until DONE is non-zero:

定义宏 TESTLOOP，其中循环调用宏命令 TEST 直到 DONE 为非零：

```
MACRO TESTLOOP
.TEST
IF {DONE}=0 THEN EXIT TESTLOOP
/
```

Defining a natural logarithm function usable within braced expressions:

定义可在大括号表达式中使用的自然对数函数：

```
MACRO LN /FUN
    SET LN = LOG %1 DIV LOG 2.718281828
/
```

Listing all macro names:

显示所有宏的名称：

```
MACRO
```