

命令模式

SET variable1 = expression1 [, variable2 = expression2]...

Sets the named variable(s) to the value(s) of the given expression(s).

给变量赋值或公式。

参数说明

variable

The name of a variable that (1) was declared by the VARIABLES command; (2) is one of the predefined system variables that allow its value to be set; or (3) is an environment variable enclosed in "%" chars (effective for the current program and any SHELLED subprograms).

通过下列方式定义的变量名：

- 1) 通过命令 VARIABLES 定义的变量。
- 2) 可以赋值的提前定义的系统变量。
- 3) 用%包括的环境变量（当前程序或子程序的有效变量）。

expression

A value, or an operator followed by an expression, or an expression followed by an operator followed by an expression.

数值，或运算公式。

value

A number or character string (which should be enclosed in quotation marks if multiple word).

数值或字符串（如果多于一个单词应该用引号引起）。

operator

Any of the following keywords (only the first two letters are significant if no conflict):

下面的任何一个关键词（每个关键词的前两个没有重复的字母都是有效的）。

Keyword Meaning

PLUS: Addition (same as ADD) 相加

MINUS: Subtraction or negation (same as SUBTRACT) 相减或负的（和命令 SUBTRACT 相同）

TIMES: Multiplication (same as MULTIPLY) 相乘

DIVIDE: Division 相除

POWER: Raises to the following power 乘方

LOG: Base 10 logarithm 10 的对数

SIN: Sine (argument in degrees) 正弦

COS:	Cosine	余弦
TAN:	Tangent	正切
ASIN:	Arcsine (result from -90 to 90 degrees)	反正弦
ACOS:	Arccosine (result from 0 to 180 degrees)	反余弦
ATAN:	Arctangent (result from -90 to 90 degrees)	反正切
ABS:	Absolute value	绝对值
SQRT:	Square root	开方根
TRUNC:	Floors to the largest integer not greater than itself	小于本身的最大整数
CHARS:	Number of characters in the following string	下面字符串中的字符个数
CHR n:	Character having ASCII value n	具有 ASCII 值 n 的字符
LEFT n:	Leftmost n characters of the following string	下面字符串中最左端的 n 个字符
RIGHT n:	Rightmost n characters of the following string	下面字符串中最右端的 n 个字符
SLICE n:	Slice of the following string starting at the nth character	从第 n 个字符将线面的字符串分开
ITEM n:	Extracts nth list item separated by spaces, tabs, or commas	将用空格, TAB 键或逗号分开的第 n 项提取出来
NEXT 0:	Position (0 if none) of any 1st string char found in 2nd string	在第二个字符串中找到的任何第一个字符串字符的位置 (如果没有, 则为 0)
NEXT n:	Position where next found 1st string in 2nd string starting at n	下一个在第二个字符串中找到第一个字符串的位置, 从 n 开始
NEXT -n:	Position next case-insensitively found 1st in 2nd starting at n	下一个不区分大小写, 从 n 开始, 在第二个中找到第 1 个
PHASE:	All characters before any finding of 1st string in 2nd string	在第二个字符串中找到第一个字符串之前的所有字符
FRNAME n:	Nth frame name, or empty if no such frame	第 n 个肋位名称, 如果没有此类帧, 则为空
FRLOC:	Frame location of the next string, or undefined if no such frame	下一个字符串的肋位位置, 如果没有这样的肋位, 则未定义
CAP:	Capitalizes the next string	将下列的字符串大写
FILENAME:	Extracts the filename and extension from a file specification	从指定文件中提取出文件名和扩展名
CLEAN:	Cleans surrounding blanks and non-printables from next string	将字符串前后的空格删除, 也不打印输出

STR: Cleans the rest of the line, translating quotes to apostrophes 将字符串后的空行删除，将引用转变成单引号内的内容

QUOTE: Takes the entire rest of the line verbatim as a string 将余下的空行也作字符串处理

Note: An acceptable substitute for SET variable = "{expression}" is:

注意: SET 变量= “{表达式}” 的可接受替代品是:

variable := expression

for assigning PEMDAS expressions to variables (see the VARIABLE command). All nonarithmetic operators useful for functions (i.e. LOG through CLEAN) provide their keywords as names for predefined functions (see the MACRO command).

用于将 PEMDAS 表达式分配给变量（请参阅变量命令）。所有对函数有用的非算术运算符（即.LOG 通过 CLEAN）都提供其关键字作为预定义函数的名称（请参阅 MACRO 命令）。

Operation

操作

The value of each expression is stored under the corresponding variable name as a character string, even if the variable type is REAL. No range checking is done.

每个描述的值以字符串的型式保存在相应的变量名下，即使变量的类型是实数。并不校核数值的范围。

Binary operations (such as PLUS) are evaluated strictly left-to-right, and no parentheses are allowed. Unary operations (such as SQRT) can be nested and operate from the inside out. All variable substitutions are performed before any of the expressions are evaluated. The final expression value is rounded to 5 decimal places (or as specified by the SPECIAL SYSDEC command) with trailing decimal zeroes trimmed.

二元计算（如相加）直接从左向右计算，不允许含有括号。一元计算（如开方根）可以从里向外的嵌套。在任何的表达式计算前，所有的变量都可以替代。最终表达式值四舍五入到小数点后 5 位（或由 SPECIAL SYSDEC 命令指定），并修剪尾随的小数零。

Retrieval of string variables containing quote marks (") is handled transparently even when the variable in braces is itself enclosed in quotes (e.g. "{X}"). Literal quote marks may be stored in a variable either by surrounding everything in quotes and doubling each internal quote mark (e.g. SET X=""text"") or by using the QUOTE function (e.g. SET X=QUOTE "text").

引号在调用过程中，会被忽略，当用引号引起带有大括号的变量时（例如：“{X}”），引号同样会被忽略。在变量中储存引号的方法是，用两个引号将引号引起（例如：SET X=""text"”，则 X 为 "text"），或者通过关键字 QUOTE 来实现（例如：SET X=QUOTE "text"）。

Invalid operations (such as division by zero or non-numeric arithmetic) set variables to "Undefined". Note that UNDEFINED matches the empty string in IF comparisons, so invalid results can be tested using IF {X}="" THEN statements.

无效的计算（如被 0 除或非数值计算）使变量无效。在 IF 语句中 UNDEFINED 与空格相匹配，因此无效的结果可以用语句 IF {X}="" THEN 来测定。

Output:

输出

none.

无

Examples

样例

Using a variable to avoid putting the same number in two places:

利用变量避免在两个地方赋予相同的值：

```
SET WGT = 1200
\*** {WGT}-TON CRANE LOAD ***\
ADD "CRANE LOAD" {WGT} * * *
```

Setting ERROR to avoid terminating when an error occurs:

定义 ERROR 避免发生错误时结束程序计算

SET ERROR = -1 `This will prevent a bad TANKNAME from being fatal. 避免错误的舱室名称而停止运算。

```
TYPE {TANKNAME} FLOODED
```

IF {ERROR} <> -1 THEN WAIT `Wait for operator to read any error message. 等待操作者读取错误信息。

SET ERROR = 0 `Restore . ERROR to its normal value. 恢复 ERROR 为正常值。

Setting variables to expression values:

将变量设置为表达式值：

```
SET N = {N} PLUS 1
```

```
SET M = MINUS {M}
```

```
SET INITIAL = LEFT 1 "{NAME}"
```

SET A = 3 PLUS 1 DIV 2 `The answer is 2, not 3.5 结果为 2，不是 3.5。

SET X = 2 PLUS SQRT ABS -16 TIMES CHARS "ABC" `The answer is 18 结果为 18

```
SET TLIST="hold1.c, hold2.c" | SET POS = NEXT 0 " , " "{TLIST}"
```

SET POS = NEXT -{POS} "HOLD" "{TLIST}" `find next hold after separator 在分隔符之后查找下一个保留

Assigning the same values using SET-free expressions and functions:

使用无 SET 表达式和函数分配相同的值:

```
N := N+1
```

```
M := -M
```

```
INITIAL := LEFT(1,"{NAME}")
```

```
A := (3+1)/2 `The answer is 2, whereas A:=3+1/2 assigns 3.5
```

```
A := (2 + SQRT(ABS(-16))) * CHARS("ABC") `The answer is 18
```

```
TLIST := "hold1.c, hold2.c" | POS := NEXT(0," ", "{TLIST}")
```

```
POS := NEXT(-{POS}, "HOLD", "{TLIST}") `find next hold after separator
```

Parsing contents and any temperature after "@" for current tank:

解析当前储罐的内容和 "@" 之后的任何温度:

```
SET CONTENTS = PARSE "@" "{TCONDESC}"
```

```
SET N = CHARS "{CONTENTS}" PLUS 2
```

```
SET CONTENTS = CLEAN "{CONTENTS}"
```

```
SET TEMPERATURE = CLEAN SLICE {N} "{TCONDESC}"
```

Assigning the DISTANCE between points (X1,Y1,Z1) and (X2,Y2,Z2):

分配点 (X1, Y1, Z1) 和 (X2, Y2, Z2) 之间的距离:

```
DISTANCE := ((X2-X1)^2 + (Y2-Y1)^2 + (Z2-Z1)^2)^(1/2)
```