

User-Defined Constants

Constants can be used throughout Natural programs. This document discusses the types of constants that are supported and how they are used.

The following topics are covered:

- Numeric Constants
 - Alphanumeric Constants
 - Date and Time Constants
 - Hexadecimal Constants
 - Logical Constants
 - Floating Point Constants
 - Attribute Constants
 - Defining Named Constants
-

Numeric Constants

A numeric constant may contain 1 to 29 numeric digits.

A numeric constant used with a COMPUTE, MOVE, or arithmetic statement may contain a decimal point and sign notation.

Examples:

```
MOVE 3 TO #XYZ
COMPUTE #PRICE = 23.34
COMPUTE #XYZ = -103
COMPUTE #A = #B * 6074
```

Alphanumeric Constants

An alphanumeric constant may contain 1 to 253 alphanumeric characters.

An alphanumeric constant must be enclosed in either apostrophes (') or quotation marks (").

Examples:

```
MOVE 'ABC' TO #XYZ
MOVE '% INCREASE' TO #TITLE
DISPLAY "LAST-NAME" NAME
```

If you want an apostrophe to be part of an alphanumeric constant that is enclosed in apostrophes, you must write this as two apostrophes or as a single quotation mark.

If you want an apostrophe to be part of an alphanumeric constant that is enclosed in quotation marks, you write this as a single apostrophe.

Example:

If you want the following to be output:

```
HE SAID, 'HELLO'
```

you can use any of the following notations:

```
WRITE 'HE SAID, ' 'HELLO' ' '
WRITE 'HE SAID, "HELLO" '
WRITE "HE SAID, " "HELLO" " "
WRITE "HE SAID, 'HELLO' "
```

An alphanumeric constant that is used to assign a value to a user-defined variable must not be split between statement lines.

Alphanumeric constants may be concatenated to form a single value by use of a hyphen.

Examples:

```
MOVE 'XXXXXX' -
      'YYYYYY' TO #FIELD

MOVE "ABC" - 'DEF' TO #FIELD
```

In this way, alphanumeric constants can also be concatenated with hexadecimal constants.

Date and Time Constants

A date constant may be used in conjunction with a format D variable. Date constants may have the following formats:

D' <i>yyyy-mm-dd</i> '	International date format
D' <i>dd.mm.yyyy</i> '	German date format
D' <i>dd/mm/yyyy</i> '	European date format
D' <i>mm/dd/yyyy</i> '	USA date format

where *dd* represent the number of the day, *mm* the number of the month and *yyyy* the year.

Example:

```
DEFINE DATA LOCAL
1 #DATE (D)
END-DEFINE
...
MOVE D'1997-08-11' TO #DATE
...
```

The default date format is controlled by the profile parameter DTFORM as set by the Natural administrator.

A time constant may be used in conjunction with a format T variable. A time constant has the following format:

T'*hh:ii:ss***'**

where *hh* represents hours, *ii* minutes and *ss* seconds.

Example:

```
DEFINE DATA LOCAL
1 #TIME (T)
END-DEFINE
...
MOVE T'11:33:00' TO #TIME
```

Hexadecimal Constants

A hexadecimal constant may be used to enter a value which cannot be entered as a standard keyboard character.

A hexadecimal constant is prefixed with an "H". The constant itself must be enclosed in apostrophes and may consist of the hexadecimal characters 0 - 9, A - F. Two hexadecimal characters are required to represent one byte of data.

The hexadecimal representation of a character varies, depending on whether your computer uses an ASCII or EBCDIC character set. Wenn you transfer hexadecimal constants to another computer, you may therefore have to convert the characters.

ASCII Examples:

```
H'313233'      (equivalent to the alphanumeric constant '123')
H'414243'      (equivalent to the alphanumeric constant 'ABC')
```

EBCDIC Examples:

```
H'F1F2F3'      (equivalent to the alphanumeric constant '123')
H'C1C2C3'      (equivalent to the alphanumeric constant 'ABC')
```

Hexadecimal constants may be concatenated by using a hyphen between the constants.

ASCII Example:

```
H'414243' - H'444546'  (equivalent to 'ABCDEF')
```

EBCDIC Example:

```
H'C1C2C3' - H'C4C5C6'  (equivalent to 'ABCDEF')
```

Logical Constants

The logical constants "TRUE" and "FALSE" may be used to assign a logical value to a field defined with format L.

Example:

```
DEFINE DATA LOCAL
1 #FLAG (L)
END-DEFINE
...
MOVE TRUE TO #FLAG
...
IF #FLAG ...
    statement ...
    MOVE FALSE TO #FLAG
END-IF
...
```

Floating Point Constants

Floating point constants can be used with variables defined with format F.

Example:

```

DEFINE DATA LOCAL
1 #FLT1 (F4)
END-DEFINE
...
COMPUTE #FLT1 = -5.34E+2
...

```

Attribute Constants

Attribute constants can be used with variables defined with format C (control variables). This type of constant must be enclosed within parentheses.

The following attributes may be used:

AD=D	default	CD=BL	blue
AD=B	blinking	CD=GR	green
AD=I	intensified	CD=NE	neutral
AD=N	non-display	CD=PI	pink
AD=V	reverse video	CD=RE	red
AD=U	underlined	CD=TU	turquoise
AD=C	cursive/italic	CD=YE	yellow
AD=Y	dynamic attribute		
AD=P	protected		

See also session parameters AD and CD.

Example:

```

DEFINE DATA LOCAL
1 #ATTR (C)
1 #FIELD (A10)
END-DEFINE
...
MOVE (AD=I CD=BL) TO #ATTR
...
INPUT #FIELD (CV=#ATTR)
...

```

Defining Named Constants

If you need to use the same constant value several times in a program, you can reduce the maintenance effort by defining a named constant: you define a field in the DEFINE DATA statement, assign a constant value to it, and use the field name in the program instead of the constant value. Thus, when the value has to be changed, you only have to change it once in the DEFINE DATA statement and not everywhere in the program where it occurs.

You specify the constant value in angle brackets with the keyword "CONSTANT" after the field definition in the DEFINE DATA statement. If the value is alphanumeric, it must be enclosed in apostrophes.

Example:

```
DEFINE DATA LOCAL  
1 #FIELD A (N3) CONSTANT <100>  
1 #FIELD B (A5) CONSTANT <'ABCDE'>  
END-DEFINE  
...
```

During the execution of the program, the value of such a named constant cannot be modified.