

WRITE

Syntax 1 - Dynamic Formatting

WRITE [(rep)] [NOTITLE] [NOHDR]

[(*statement-parameters*)]

$$\left\{ \begin{bmatrix} nX \\ nT \\ x/y \\ T^* \text{field-name} \\ P^* \text{field-name} \\ / \end{bmatrix} \dots \right. \left. \begin{cases} \text{'text'} \text{ [(attributes)]} \\ \text{'c'(n)} \text{ [(attributes)]} \\ [=] \text{ operand1 [(parameters)]} \end{cases} \right\} \dots$$

| Operand | Possible Structure | | | | | | | | | | Possible Formats | | | | | Referencing Permitted | Dynamic Definition |
|----------|-------------------------------|--|--|--|--|--|--|--|--|--|------------------|--|--|--|--|-----------------------|--------------------|
| Operand1 | S A G N A N P I F B D T L G O | | | | | | | | | | | | | | | yes | no |

Related Statements: DISPLAY | WRITE TITLE | WRITE TRAILER | INPUT

See also Output of Data, Statements DISPLAY and WRITE in the Natural Programming Guide.

Function

The WRITE statement is used to produce output in free format.

The WRITE statement differs from the DISPLAY statement in the following respects:

- Line overflow is supported. If the line width is exceeded for a line, the next field (or text) is written on the next line. Fields or text elements are not split between lines.
- No default column headers are created. The length of the data determines the number of positions printed for each field.
- A range of values/occurrences for an array is output horizontally rather than vertically.

Report Specification - rep

The notation (*rep*) is used to specify the number of the report if multiple reports are to be produced by the program. A value in the range 0 - 31 or a logical name which has been assigned using the DEFINE PRINTER statement may be specified. If (*rep*) is not specified, the statement will apply to the first report (report 0).

If this printer file is defined to Natural as PC, the report will be downloaded to the PC.

NOTITLE

Natural generates a single title line for each page resulting from a WRITE statement. This title contains the page number, the time of day, and the date. Time of day is set at the beginning of program execution.

This title line may be overridden by using a WRITE TITLE statement, or it may be suppressed by specifying the NOTITLE clause in the WRITE statement.

Examples:

```
Default title will be produced:  
WRITE NAME
```

```
User title will be produced:  
WRITE NAME  
WRITE TITLE 'USER TITLE'
```

```
No title will be produced:  
WRITE NOTITLE NAME
```

If the NOTITLE option is used, it applies to all DISPLAY, PRINT and WRITE statements within the same object which write data to the same report.

Page overflow is checked **before** execution of a WRITE statement. No new page with title or trailer information is generated **during** the execution of a WRITE statement.

NOHDR

The WRITE statement itself does not produce any column headers. However, if you use the WRITE statement in conjunction with a DISPLAY statement, you can use the NOHDR option of the WRITE statement to suppress the column headers generated by the DISPLAY statement: the NOHDR option only takes effect if the WRITE statement is executed **after** a DISPLAY statement, the output spans more than one page, and the execution of the WRITE statement causes a new page to be output. Without the NOHDR option, the column headers of the DISPLAY statement would be output on this new page; with NOHDR they will not.

statement-parameters

One or more parameters, enclosed within parentheses, may be specified immediately after the WRITE statement (see table and example below).

Each parameter specified will override any previous parameter specified in a GLOBALS command, SET GLOBALS or FORMAT statement. If more than one parameter is specified, they must be separated by one or more blanks from one another. Each parameter specification must not be split between two statement lines.

The parameter settings applied here will only be regarded for variable fields, but they have no effect on text-constants. If you would like to set field attributes for a text-constant, they have to be set explicitly for this element.

| Parameters that can be specified with the WRITE statement | | Specification S = at statement level E = at element level |
|--|-------------------------------------|--|
| AD | Attribute Definition | SE |
| AL | Alphanumeric Length for Output | SE |
| BX | Box Definition | SE |
| CD | Color Definition | SE |
| CV | Control Variable | SE |
| DF | Date Format | SE |
| DY | Dynamic Attributes | SE |
| EM | Edit Mask | SE |
| FL | Floating Point Mantissa Length | SE |
| IS | Identical Suppress | SE |
| LS | Line Size | S |
| MC | Multiple-Value Field Count | S |
| MP | Maximum Number of Pages of a Report | S |
| NL | Numeric Length for Output | SE |
| PC | Periodic Group Count | S |
| PM | Print Mode | SE |
| PS | Page Size | S |
| SG | Sign Position | SE |
| UC | Underlining Character | S |
| ZP | Zero Printing | SE |

The individual parameters are described in the section Session Parameters of the Natural Parameter Reference documentation.

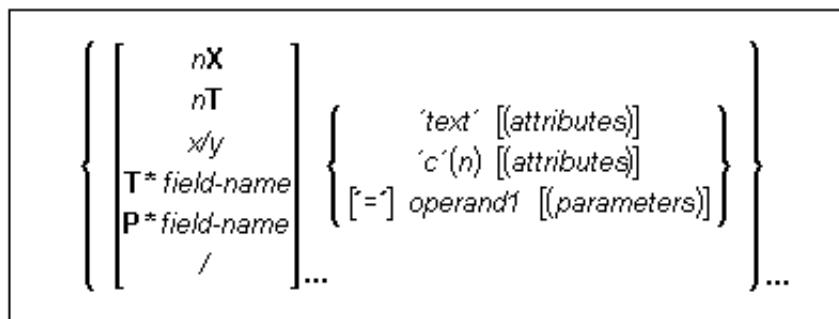
Example:

```

DEFINE DATA LOCAL
1 VARI (A4)      INIT <'1234'>          /* Output
END-DEFINE          /* Produced
*
/* -----
WRITE      'Text'           VARI          /* Text 1234
WRITE (PM=I) 'Text'           VARI          /* Text 4321
WRITE      'Text' (PM=I)     VARI (PM=I)  /* txeT 4321
WRITE      'Text' (PM=I)     VARI          /* txeT 1234
END

```

Output Format



Field Positioning Notations

nX

Note: (for Mainframes Only)

This notation inserts n spaces between columns. n must not be "0".

Example: WRITE NAME 5X SALARY

nT

The nT notation causes positioning (tabulation) to print position "n". Backward positioning is not permitted.

Example: WRITE 25T NAME 50T SALARY

(causes NAME to print beginning in position 25 and SALARY to print beginning in position 50).

x/y

Causes the next element to be placed x lines below the output of the last statement, beginning in column y. y must not be "0". Backward positioning in the same line is not permitted.

T*field-name

The notation T* is used to position to a specific print position of a field used in a previous DISPLAY statement. Backward positioning is not permitted.

P**field-name*

The notation P* is used to position to a specific print position *and line* of a field used in a previous DISPLAY statement. It is most often used in conjunction with vertical printing mode. Backward positioning is not permitted.

Equal Sign '='

When placed before a field, '=' results in the display of the field heading (as defined in the DEFINE DATA statement or in the DDM) followed by the field contents.

Slash '/'

When placed between fields or text elements, "/" causes positioning to the beginning of the next print line.

Example: WRITE NAME / SALARY

Multiple "/" notations may be used to cause multiple line advances.

Text/Attribute Assignment

'text'

text is displayed.

Example: WRITE 'EMPLOYEE' NAME 'MARITAL/STATUS' MAR-STAT

'c'(n)

Identical to 'text' except that the specified character *c* is displayed *n* times.

Example: WRITE '*' (5) '=' NAME

attributes

Indicates the display and color attributes to be used for text/field display. The following *attributes* can be used:

| | |
|-----|------|
| [B] | [BL] |
| C | GR |
| D | NE |
| I | PI |
| N | RE |
| U | TU |
| V | YE |
| 1 | 2 |

1. Display attributes (see the session parameter AD in the Natural Parameter Reference documentation).
2. Color attributes (see the session parameter CD in the Natural Parameter Reference documentation).

WRITE 'TEXT' (BGR)

WRITE 'TEXT' (B)

WRITE 'TEXT' (BBLC)

operand1

The field to be written.

Note for DL/I databases:

The DL/I AIX fields can be displayed only if a PCB is used with the AIX specified in the parameter PROCSEQ. If not, an error message is returned by Natural at runtime.

parameters

One or more parameters, enclosed within parentheses, may be specified immediately after *operand1*. Each parameter specified in this manner will override any previous parameter specified in a GLOBALS command, SET GLOBALS or FORMAT statement. If more than one parameter is specified, one or more blanks must be present between each entry. An entry may not be split between two statement lines.

For information on the individual parameters, see the table in the section statement-parameters.

Syntax 2 - Using Predefined Map

| | | | | | | | |
|-------|------------------|-----------|---------|---------|---------------|-----------------|------------------------|
| WRITE | [(<i>rep</i>)] | [NOTITLE] | [NOHDR] | [USING] | {FORM MAP} | <i>operand1</i> | [<i>operand2</i> ...] |
|-------|------------------|-----------|---------|---------|---------------|-----------------|------------------------|

| Operand | Possible Structure | Possible Formats | | | | | | | | | | | | Referencing Permitted | Dynamic Definition | |
|----------|--------------------|------------------|---|---|---|---|---|---|---|---|---|---|---|-----------------------|--------------------|----|
| Operand1 | C S | | | | A | | | | | | | | | no | no | |
| Operand2 | | S | A | G | N | A | N | P | I | F | B | D | T | L | yes | no |

FORM/MAP

This option may be used to indicate that a form/map layout previously defined using the Natural map editor is to be used.

A map layout used in a WRITE statement does not automatically create a new page each time the map is output.

The LS parameter setting must be 1 byte greater than the LS setting defined in the map.

operand1

The name of the form/map to be used.

operand2

The field to be written.

If *operand1* is a constant and *operand2* is omitted, the fields are taken from the map source at compilation time.

WRITE

NOTITLE/NOHDR

NOTITLE/NOHDR

NOTITLE and NOHDR are described under Syntax 1 of the WRITE statement.

Example 1

```
/* EXAMPLE 'WRTEX1': WRITE (USING '=', 'TEXT', '/')  
*****  
LIMIT 1  
READ EMPLOYEES BY NAME  
*****  
WRITE NOTITLE '=' NAME '=' FIRST-NAME '=' MIDDLE-I //  
    'LOCATION' /  
    'CITY:' CITY /  
    'COUNTRY:' COUNTRY //  
*****  
END
```

| | | |
|---------------|------------------|-----------|
| NAME: ABELLAN | FIRST-NAME: KEPA | MIDDLE-I: |
| LOCATION | | |
| CITY: MADRID | | |
| COUNTRY: E | | |

Example 2

```
/* EXAMPLE 'WRTEX2': WRITE (USING NX, NT NOTATION)  
*****  
LIMIT 4  
READ EMPLOYEES BY NAME  
WRITE NOTITLE 5X NAME 50T JOB-TITLE  
*****  
END
```

| | |
|----------|-------------------------|
| ABELLAN | MAQUINISTA |
| ACHIESON | DATA BASE ADMINISTRATOR |
| ADAM | CHEF DE SERVICE |
| ADKINSON | SALES PERSON |

Example 3

```
/* EXAMPLE 'WRTEX3': WRITE (USING T* NOTATION)
*****
LIMIT 5
READ EMPLOYEES BY CITY STARTING FROM 'ALBU'
DISPLAY NOTITLE CITY NAME SALARY (1)
AT BREAK CITY
*****
WRITE / 'CITY AVERAGE:' T*SALARY (1) AVER(SALARY(1)) //
*****
END
```

| CITY | NAME | ANNUAL SALARY |
|---------------|----------|------------------|
| <hr/> | | |
| ALBUQUERQUE | HAMMOND | 22000 |
| ALBUQUERQUE | ROLLING | 34000 |
| ALBUQUERQUE | FREEMAN | 34000 |
| ALBUQUERQUE | LINCOLN | 41000 |
| CITY AVERAGE: | | 32750 |
| ALFRETON | GOLDBERG | 4700 |
| CITY AVERAGE: | | 4700 |

Example 4

```
* EXAMPLE 'WRTEX4': WRITE (USING P* NOTATION)
*****
LIMIT 3
READ EMPLOYEES BY CITY FROM 'N'
DISPLAY NOTITLE NAME CITY
    VERT AS 'BIRTH/SALARY' BIRTH (EM=YYYY-MM-DD) SALARY (1)
SKIP 1
AT BREAK CITY
*
WRITE / 'CITY AVERAGE:' P*SALARY (1) AVER(SALARY(1)) //
*
LOOP
END
```

| NAME | CITY | BIRTH SALARY |
|--------------|-----------|----------------------|
| WILCOX | NASHVILLE | 1970-01-01 38000 |
| MORRISON | NASHVILLE | 1949-07-10 36000 |
| CITY AVERAGE | | 37000 |
| BOYER | NEMOURS | 1955-11-23 195900 |
| CITY AVERAGE | | 195900 |

Example 5

```
/* EXAMPLE 'WRTEX5': WRITE (USING '=', STATEMENT/ELEMENT PARAMETERS)
*****
LIMIT 2
READ EMPLOYEES BY NAME
  WRITE NOTITLE (AL=16 NL=8)
    '=' PERSONNEL-ID '=' NAME '=' PHONE (AL=10 EM=XXX-XXXXXXX)
*****
END
```

| | | |
|------------------------|----------------|--------------------|
| PERSONNEL ID: 60008339 | NAME: ABELLAN | TELEPHONE: 435-672 |
| PERSONNEL ID: 30000231 | NAME: ACHIESON | TELEPHONE: 523-341 |