WRITE

This chapter covers the following topics:

- Function
- Syntax 1 - Dynamic Formatting
- Syntax 1 - Description
- Syntax 2 - Using Predefined Form/Map
- Syntax 2 - Description
- Examples

For an explanation of the symbols used in the syntax diagram, see *Syntax Symbols*.

Related Statements: AT END OF PAGE | AT TOP OF PAGE | CLOSE PRINTER | DEFINE PRINTER | DISPLAY | EJECT | FORMAT | NEWPAGE | PRINT | SKIP | SUSPEND IDENTICAL SUPPRESS | WRITE TITLE | WRITE TRAILER

Belongs to Function Group: *Creation of Output Reports*

Function

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

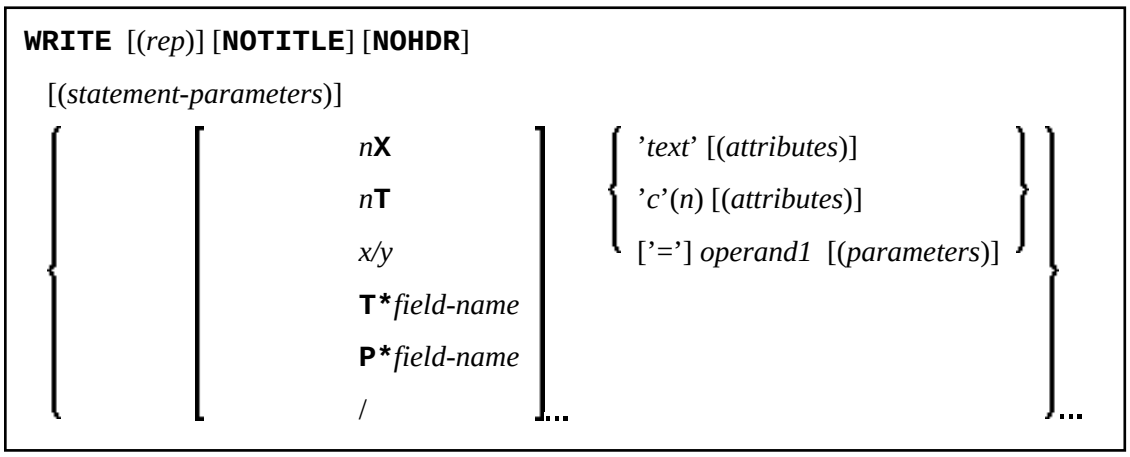
The `WRITE` statement differs from the `DISPLAY` statement in the following respects:

- 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.

See also the following topics (in the *Programming Guide*):

- *Controlling Data Output*
- *Statements DISPLAY and WRITE*
- *Index Notation for Multiple-Value Fields and Periodic Groups*
- *Example of DISPLAY VERT with WRITE Statement*
- *Layout of an Output Page*

Syntax 1 - Dynamic Formatting



For an explanation of the symbols used in the syntax diagram, see *Syntax Symbols*.

Syntax 1 - Description

Operand Definition Table:

Operand	Possible Structure					Possible Formats													Referencing Permitted	Dynamic Definition
operand1		S	A	G	N	A	U	N	P	I	F	B	D	T	L		G	O	yes	no

Syntax Element Description:

(*rep*)	**Report Specification:** The notation `(rep)` is used to specify the identification of the report if multiple reports are to be produced by the program. As report identification, 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, see *Example 5*. For information on how to control the format of an output report created with Natural, see *Controlling Data Output* (in the *Programming Guide*).

NOTITLE	**Default Page Title Suppression:**
	
	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 default title line may be overridden by using a `WRITE TITLE` statement, or it may be suppressed by using the `NOTITLE` option 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`
	
	Notes:
	
	1. 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.
	2. 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	**Column Header Suppression:** 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 execution of the `WRITE` statement causes a new page to be output. Without the `NOHDR` option, the column headers (if any) of the `DISPLAY` statement would be output on this new page; with `NOHDR` they will not.

statement-parameters	**Parameter Definition at Statement Level:** One or more parameters, enclosed within parentheses, may be specified at statement level, that is, immediately after the `WRITE` statement. Each parameter specified will override the corresponding parameter previously specified in a `GLOBALS` command, `SET GLOBALS` (in Reporting Mode only) 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. **Note:** 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, see *Parameter Definition at Element (Field) Level*. See also: ● *List of Parameters* ● *Example of Parameter Usage at Statement and Element (Field) Level* ● *Example 5 - WRITE Statement Using '=' and Parameters on Statement/Element (Field) Level*
***n*X, *n*T, *x/y*, T**field-name*, P**field-name*, '=', /,**	**Field Positioning Notation:** See *Field Positioning Notations* in the section *Output Format Definitions*.
'*text*', '*c*'(*n*), *attributes*, *operand1*, *parameters*	**Text/Attribute Assignment:** See *Text/Attribute Assignment* in the section *Output Format Definitions*.

List of Parameters

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
DL	Display Length for Output	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

* If the number of occurrences of an array exceeds the PS value, a NAT0303 error will be output.

The individual session parameters are described in the *Parameter Reference*.

See also the following topics in the *Programming Guide*:

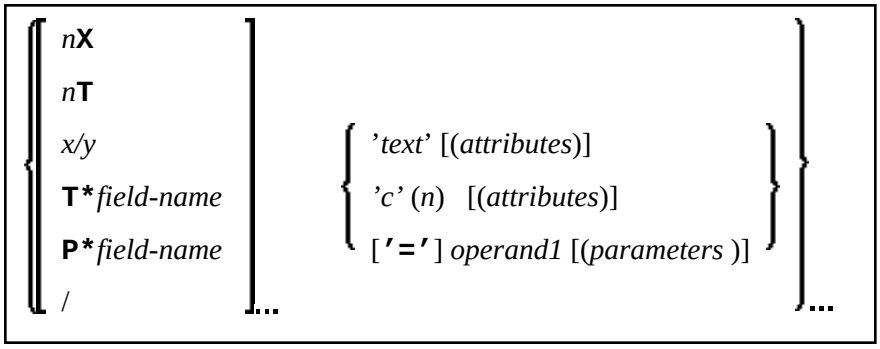
- *Centering of Column Headers - HC Parameter*
- *Width of Column Headers - HW Parameter*
- *Filler Characters for Headers - Parameters FC and GC*
- *Underlining Character for Titles and Headers - UC Parameter*

Example of Parameter Usage at Statement and Element (Field) Level

```
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
```

See also *Example 5 - WRITE Statement Using '=' and Parameters on Statement/Element (Field) Level*.

Output Format Definitions



For an explanation of the symbols used in the syntax diagram, see *Syntax Symbols*.

Field Positioning Notations

***n*X**	**Column Spacing:** This notation inserts *n* spaces between columns. *n* must not be zero. Example: `WRITE NAME 5X SALARY` See also: • *Example 2 - WRITE Statement Using nX, nT Notation* (below) • *Column Spacing - SF Parameter and nX Notation* (in the *Programming Guide*)

***n*T**	**Tab Setting:** The *n*T notation causes positioning (tabulation) to print position *n*. Backward positioning is not permitted. In the following example, NAME is printed beginning in position 25, and SALARY is printed beginning in position 50: `WRITE 25T NAME 50T SALARY` See also: ● *Example 2 - WRITE Statement Using nX, nT Notation* (below) ● *Tab Setting - nT Notation* (in the *Programming Guide*)
x/y	***x/y* Positioning:** The *x*/*y* notation causes the next element to be placed *x* lines below the output of the last statement, beginning in column *y*. *y* must not be zero. Backward positioning in the same line is not permitted. See also *Positioning Notation x/y* (in the *Programming Guide*).
T**field-name***	**Field Related Positioning:** 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. See also: ● *Example 3 - WRITE Statement Using T* Notation* (below) ● *Tab Notation - T*field* (in the *Programming Guide*)
P**field-name***	**Field and Line Related Positioning:** 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. See also: ● *Example 4 - WRITE Statement Using P* Notation* (below) ● *Tab Notation P*field* (in the *Programming Guide*)

'='	**Field Content Positioned behind Field Heading:** When placed before a field, the equal sign `'='` results in the display of the field heading (as defined in the `DEFINE DATA` statement or in the DDM) followed by the field contents. See also: • *Example 1 - WRITE Statement Using '=', 'text', '/'* • *Example 5 - WRITE Statement Using '=' and Statement/Element Parameters*
/	**Line Advance - Slash Notation:** When placed between fields or text elements, a slash (/) causes positioning to the beginning of the next print line. Example: `WRITE NAME / SALARY` Multiple slash (/) notations may be used to cause multiple line advances. See also: • *Example 1 - WRITE Statement Using '=', 'text', '/'* (below) • *Line Advance - Slash Notation* (in the *Programming Guide*) • *Example 2 - Line Advance in WRITE Statement* (in the *Programming Guide*)

Text/Attribute Assignment

'text'	**Text Assignment:** The character string enclosed by single quotes is displayed. Example: `WRITE 'EMPLOYEE' NAME 'MARITAL/STATUS' MAR-STAT` See also: • *Example 1 - WRITE Statement Using '=', 'text', '/'* (below) • *Text Notation, Defining a Text to Be Used with a Statement* (in the *Programming Guide*)

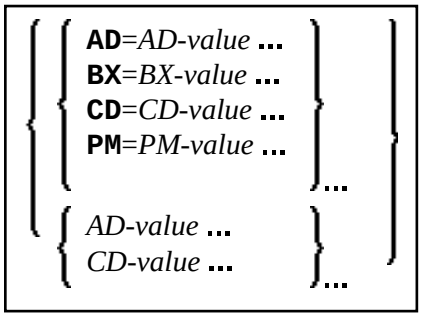
'c'(n)	**Character Repetition:**
	
	The character enclosed by single quotes is displayed *n* times immediately before the field value.
	
	For example:
	
	`WRITE '*' (5) '=' NAME`
	
	results in
	
	`***** SMITH`
	
	See also *Text Notation, Defining a Character to Be Displayed n Times before a Field Value* (in the *Programming Guide*).

attributes	**Field Representation and Color Attributes:**
	
	It is possible to assign various attributes for text/field display. These attributes and the syntax that may be used are described in the section *Output Attributes* below.
	
	Examples:
	
	`WRITE 'TEXT' (BGR)`
	`WRITE 'TEXT' (B)`
	`WRITE 'TEXT' (BBLC)`

operand1	**Field to be Written:** *operand1* specifies the field whose content is to be written in this place. Arrays with ranges that allow to vary the number of occurrences at execution time may not be specified. **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	**Parameter Definition at Element (Field) Level:** One or more parameters, enclosed within parentheses, may be specified at element (field) level, that is, immediately after *operand1*. Each parameter specified in this manner will override the corresponding parameter previously specified at statement level or in a `GLOBALS` command, `SET GLOBALS` (in Reporting Mode only) or `FORMAT` statement. If more than one parameter is specified, one or more blanks must be placed between each entry. An entry may not be split between two statement lines. See also: • *List of Parameters* • *Example of Parameter Usage at Statement and Element (Field) Level*

Output Attributes

attributes indicates the output attributes to be used for text display. Attributes may be:

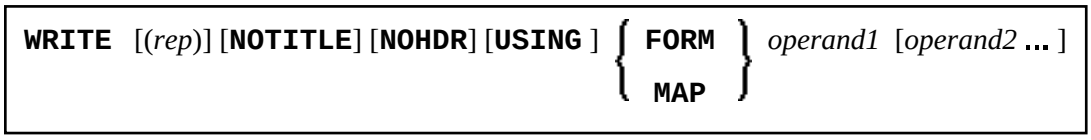


For the possible session parameter values, refer to the corresponding sections in the *Parameter Reference* documentation:

- *AD - Attribute Definition*, section *Field Representation*
- *CD - Color Definition*
- *BX - Box Definition*
- *PM - Print Mode*

Note:
The compiler actually accepts more than one attribute value for an output field. For example, you may specify: `AD=BDI`. In such a case, however, only the last value applies. In the given example, only the value `I` will become effective and the output field will be displayed intensified.

Syntax 2 - Using Predefined Form/Map



For an explanation of the symbols used in the syntax diagram, see *Syntax Symbols*.

Syntax 2 - Description

Operand Definition Table:

Operand	Possible Structure					Possible Formats												Referencing Permitted	Dynamic Definition
operand1	C	S				A												no	no
operand2		S	A	G	N	A	U	N	P	I	F	B	D	T	L			yes	no

Syntax Element Description:

[USING] FORM [USING] MAP	**Use of Predefined Form/Map Layout:** 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. For the line spacing, the `LS` parameter setting must be 1 byte greater than the `LS` setting defined in the map.
operand1	**Form/Map Name:** *operand1* is the name of the form/map to be used.
operand2	**Field to be Written:** *operand2* is the name(s) of the field(s) to be written. If *operand1* is a constant and *operand2* is omitted, the fields are taken from the map source at compilation time. The fields must agree in number, sequence, format, length and (for arrays) number of occurrences with the fields in the referenced form/map; otherwise, an error occurs.
NOTITLE/NOHDR	**Title Line/Column Header Suppression:** `NOTITLE` and `NOHDR` are described under *Syntax 1* of the `WRITE` statement.

Examples

- Example 1 - WRITE Statement Using ’=’, ’text’, ’/’
- Example 2 - WRITE Statement Using nX, nT Notation
- Example 3 - WRITE Statement Using T* Notation
- Example 4 - WRITE Statement Using P* Notation
- Example 5 - WRITE Statement Using ’=’ and Parameters on Statement/Element (Field) Level
- Example 6 - Report Specification with Output File Defined to Natural as PC

Example 1 - WRITE Statement Using ’=’, ’*text*’, ’/’

```
** Example 'WRTEX1': WRITE (with '=', 'text', '/')
************************************************************************
DEFINE DATA LOCAL
1 EMPL-VIEW VIEW OF EMPLOYEES
  2 FULL-NAME
    3 FIRST-NAME
```

```
    3 MIDDLE-I
    3 NAME
  2 CITY
  2 COUNTRY
END-DEFINE
*
LIMIT 1
READ EMPL-VIEW BY NAME
  /*
  WRITE NOTITLE
        '=' NAME '=' FIRST-NAME '=' MIDDLE-I //
        'L O C A T I O N' /
        'CITY:   ' CITY    /
        'COUNTRY:' COUNTRY //
  /*
END-READ
END
```

Output of Program WRTEX1:

```
NAME: ABELLAN              FIRST-NAME: KEPA                MIDDLE-I:

L O C A T I O N
CITY:    MADRID
COUNTRY: E
```

Example 2 - WRITE Statement Using *n*X, *n*T Notation

```
** Example 'WRTEX2': WRITE (with nX, nT notation)
************************************************************************
DEFINE DATA LOCAL
1 EMPL-VIEW VIEW OF EMPLOYEES
  2 NAME
  2 JOB-TITLE
END-DEFINE
*
LIMIT 4
READ EMPL-VIEW BY NAME
  WRITE NOTITLE 5X NAME 50T JOB-TITLE
END-READ
END
```

Output of WRTEX2:

```
ABELLAN                                          MAQUINISTA
ACHIESON                                         DATA BASE ADMINISTRATOR
ADAM                                             CHEF DE SERVICE
ADKINSON                                         PROGRAMMER
```

Example 3 - WRITE Statement Using T* Notation

```
** Example 'WRTEX3': WRITE (with T* notation)
************************************************************************
DEFINE DATA LOCAL
1 EMPL-VIEW VIEW OF EMPLOYEES
  2 NAME
  2 CITY
  2 SALARY (1)
END-DEFINE
*
LIMIT 5
```

```
READ EMPL-VIEW BY CITY STARTING FROM 'ALBU'
  DISPLAY NOTITLE CITY NAME SALARY (1)
  AT BREAK CITY
    /*
    WRITE / 'CITY AVERAGE:' T*SALARY (1) AVER(SALARY(1)) //
    /*
  END-BREAK
END-READ
END
```

Output of Program WRTEX3:

```
        CITY                 NAME           ANNUAL
                                            SALARY
-------------------- -------------------- ----------

ALBUQUERQUE          HAMMOND                   22000
ALBUQUERQUE          ROLLING                   34000
ALBUQUERQUE          FREEMAN                   34000
ALBUQUERQUE          LINCOLN                   41000

CITY AVERAGE:                                  32750


ALFRETON             GOLDBERG                   4800

CITY AVERAGE:                                   4800
```

Example 4 - WRITE Statement Using P* Notation

```
** Example 'WRTEX4': WRITE (with P* notation)
************************************************************************
DEFINE DATA LOCAL
1 EMPL-VIEW VIEW OF EMPLOYEES
  2 NAME
  2 CITY
  2 BIRTH
  2 SALARY (1)
END-DEFINE
*
LIMIT 3
READ EMPL-VIEW 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)) //
  END-BREAK
END-READ
END
```

Output of Program WRTEX4:

```
        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 - WRITE Statement Using '=' and Parameters on Statement/Element (Field) Level

```
** Example 'WRTEX5': WRITE (using '=', statement/element parameters)
************************************************************************
DEFINE DATA LOCAL
1 EMPL-VIEW VIEW OF EMPLOYEES
  2 NAME
  2 PERSONNEL-ID
  2 PHONE
END-DEFINE
*
LIMIT 2
READ EMPL-VIEW BY NAME
  WRITE NOTITLE (AL=16 NL=8)
        '=' PERSONNEL-ID '=' NAME '=' PHONE (AL=10 EM=XXX-XXXXXXX)
END-READ
END
```

Output of Program WRTEX5:

```
PERSONNEL ID: 60008339         NAME: ABELLAN          TELEPHONE: 435-6726
PERSONNEL ID: 30000231         NAME: ACHIESON         TELEPHONE: 523-341
```

Example 6 - Report Specification with Output File Defined to Natural as PC

```
** Example 'PCDIEX1': DISPLAY and WRITE to PC
**
** NOTE: Example requires that Natural Connection is installed.
************************************************************************
DEFINE DATA LOCAL
01 PERS VIEW OF EMPLOYEES
  02 PERSONNEL-ID
  02 NAME
  02 CITY
END-DEFINE
*
FIND PERS WITH CITY = 'NEW YORK'              /* Data selection
  WRITE (7) TITLE LEFT 'List of employees in New York' /
  DISPLAY (7)          /* (7) designates the output file (here the PC).
    'Location'  CITY
    'Surname'   NAME
    'ID'        PERSONNEL-ID
END-FIND
END
```