

# WRITE TRAILER

```

WRITE [(rep)] TRAILER [LEFT [JUSTIFIED]] [UNDERLINED]
  [(statement-parameters)]
  {
    [nX] { 'text' [(attributes)] }
    [nT] { 'c'(n) [(attributes)] }
    [x/y]... [=] operand1 [(parameters)] } ...
  [SKIP operand2 [LINES]]

```

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
Operand2	C	S					N	P	I	B							yes	no

Related Statements: [WRITE](#) | [DISPLAY](#) | [WRITE TITLE](#)

## Function

The WRITE TRAILER statement is used to output text or the contents of variables at the bottom of a page.

This statement is non-procedural (that is, its execution depends on an event, not on where in a program it is located).

## Restrictions

WRITE TRAILER may be specified only once per report.

WRITE TRAILER cannot be specified within a special condition statement block.

WRITE TRAILER cannot be specified within a subroutine.

## Processing

This statement is executed when an end-of-page or end-of-data condition is detected, or when a SKIP or NEWPAGE statement causes a page advance. It is not executed as a result of an EJECT statement.

The end-of-page condition is checked only after the processing of an entire DISPLAY/WRITE statement. If a DISPLAY/WRITE statement produces multiple lines of output, overflow of the physical page may occur before the end-of-page condition is reached.

### Note:

If a report is produced by statements in different objects, the WRITE TRAILER statement is only executed if it is contained in the same object as the statement that causes the end-of-page condition.

## Logical Page Size

The logical page size (specified with the session parameter PS) should be less than the physical page size to ensure that the trailer information appears at the bottom of the same page.

## Report Specification - rep

The notation (*rep*) may be used to specify the identification of the report for which the WRITE TRAILER statement is applicable. 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 WRITE TRAILER statement applies to the first report (report 0).

## Justification and Underlining

By default, the trailer lines are centered and not underlined. LEFT JUSTIFIED and UNDERLINED may be specified to override these defaults. If UNDERLINED is specified, the underlining character (either default or specified with the session parameter UC) is printed underneath the trailer and runs the width of the line size (session parameter LS).

Natural first applies all spacing or tab specifications and creates the line before centering the whole line. For example, a notation of "10T" as the first element would cause the centered header to be positioned five positions to the right.

## statement-parameters

One or more parameters, enclosed within parentheses, may be specified immediately after WRITE TRAILER. 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 a description of each parameter, see the section Session Parameters of the Natural Reference documentation.

## operand1

*Operand1* represents the field/fields to be output as trailer information.

Format notations and spacing elements are identical to those used with the WRITE statement (see the WRITE statement for more information).

## SKIP - operand2

SKIP may be used to cause lines to be skipped immediately after the trailer line. The number of lines to be skipped (*operand2*) may be specified as a numeric constant or as the content of a numeric variable.

### Note:

SKIP after WRITE TRAILER is always interpreted as the SKIP clause of the WRITE TRAILER statement, and not as an independent statement. If you wish an independent SKIP statement after a WRITE TRAILER statement, use a semicolon (;) to separate the two statements from one another.

