DISPLAY

DISPLAY [(*rep*)] [*options*] {[/ **...**] [*output-format*] *output-element*} **...**

This chapter covers the following topics:

- Function
- Syntax Description
- Defaults Applicable for a DISPLAY Statement
- 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 EJECT | FORMAT | NEWPAGE | PRINT | SKIP | SUSPEND IDENTICAL SUPPRESS | WRITE | WRITE TITLE | WRITE TRAILER

Belongs to Function Group: *Creation of Output Reports*

Function

The `DISPLAY` statement is used to specify the fields to be output on a report in column format. A column is created for each field and a field header is placed over the column.

Note:
The statements `WRITE` and `PRINT` can be used to produce output in free (non-column) format.

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*
- *Column Headers*
- *Layout of an Output Page*

Syntax Description

(*rep*)	**Report Specification:** The notation `(rep)` may be used to specify the identification of the report for which the `DISPLAY` statement is applicable. 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 8*. For information on how to control the format of an output report created with Natural, see *Controlling Data Output* (in the *Programming Guide*).
options	**Display Options:** For details, see *Display Options* below.
output-format	**Output Format Definitions:** For details, see *Output Format Definitions* below.
/	**Line Advance - Slash Notation:** When specified within a text element, a slash (/) causes a line advance for the text displayed. When specified between output elements, it causes the output element specified by the slash (/) to be placed vertically within the same column. The header for this column will be constructed by placing the headers of the vertically displayed elements vertically above the column. See also the following topics (in the *Programming Guide*): • *Line Advance - Slash Notation* • *Example 1 - Line Advance in DISPLAY Statement* • *Suppressing Column Headers - Slash Notation*
output-element	**Output Element:** For details, see *Output Element* (below).

Display Options

[**NOTITLE**] [**NOHDR**] [[**AND**] [**GIVE**] [**SYSTEM**] **FUNCTIONS**] [(*statement-parameters*)]

Syntax Element Description:

NOTITLE	**Default Page Title Suppression:**
	
	By default, Natural generates a single title line for each page resulting from a `DISPLAY` statement. This title contains the page number, the time of day, and the date. Time of day is set at the beginning of the program execution (TP mode) or at the beginning of the job (batch mode). The default title line may be overridden by using a `WRITE TITLE` statement, or it may be suppressed by specifying the keyword `NOTITLE` in the `DISPLAY` statement.
	
	Examples:
	
	• Default title will be produced:
	
	`DISPLAY NAME`
	
	• User title will be produced:
	
	`DISPLAY NAME WRITE TITLE 'user-title'`
	
	• No title will be produced:
	
	`DISPLAY NOTITLE NAME`
	
	Note:
	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.

NOHDR	**Column Headers:**
	
	Column headers are produced for each field specified in the DISPLAY statement using the following rules:
	
	● The header text may be explicitly specified in the DISPLAY statement before the field name. For example:
	
	`DISPLAY 'EMPLOYEE' NAME 'SALARY' SALARY`
	
	● If you do not specify an explicit header for a field, the header as defined in the DEFINE DATA statement will be used.
	
	● If for a database field no header is defined in the DEFINE DATA statement, the default header as defined in the DDM will be used.
	
	● If no default header is defined in the DDM, the field name will be used as header.
	
	● If for a user-defined variable no header is defined in the DEFINE DATA statement, the variable name will be used as header. See also the DEFINE DATA statement for header definition.
	
	`DISPLAY NAME SALARY #NEW-SALARY`
	
	● Natural always underlines column headings and generates one blank line between the underlining and the data being displayed.
	
	● If there are multiple DISPLAY statements in a program, the first DISPLAY statement determines the column header(s) to be used; this is evaluated at compilation time.
	
	Column Header Suppression:
	
	To suppress the column header for a single field
	
	● Specify the following characters (apostrophe-slash-apostrophe) before the field name:
	
	`'/'`
	
	For example:
	
	`DISPLAY '/' NAME 'SALARY' SALARY`
	
	To suppress all column headers
	
	● Specify the keyword NOHDR:
	
	`DISPLAY NOHDR NAME SALARY`
	
	Notes:
	
	1. NOHDR only takes effect for the first DISPLAY statement, as subsequent DISPLAY statements cannot create column headers anyhow.
	2. If both NOTITLE and NOHDR are used, they must be specified in the following order: DISPLAY NOTITLE NOHDR NAME SALARY

GIVE SYSTEM FUNCTIONS	**System Function Usage:** The `GIVE SYSTEM FUNCTIONS` clause is used to make available the following Natural system functions: `AVER`, `COUNT`, `MAX`, `MIN`, `NAVER`, `NCOUNT`, `NMIN`, `SUM`, `TOTAL`. These are evaluated when the `DISPLAY` statement containing the `GIVE SYSTEM FUNCTIONS` clause is executed. These functions may then be referred to in a statement executed as a result of an end-of-page condition. **Notes:** 1. Only one `DISPLAY` statement per report may contain a `GIVE SYSTEM FUNCTIONS` clause. When system functions are evaluated from a `DISPLAY` statement, they are evaluated on a page basis, which means that all functions (except `TOTAL`) are reset to zero when a new page is initiated. 2. When system functions are used within a `DISPLAY` statement within a subroutine, the end-of-page processing must occur within the same routine. See also *Example 2 - DISPLAY Statement Using GIVE SYSTEM FUNCTIONS Clause*.
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 `DISPLAY` statement. Each parameter specified will override the corresponding parameter previously specified in a `GLOBALS` command, `SET GLOBALS` (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 7 - DISPLAY Statement Using Parameters on Statement/Element Level

List of Parameters

Parameters that can be specified with the DISPLAY statement		**Specification (S = at statement level, E = at element level)**
AD	Attribute Definition	SE

Parameters that can be specified with the DISPLAY statement		Specification (S = at statement level, E = at element level)
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
ES	Empty Line Suppression	S
FC	Filler Character	SE
FL	Floating Point Mantissa Length	SE
GC	Filler Character for Group Headers	SE
HC	Header Centering	SE
HW	Heading Width	SE
IC	Insertion Character	SE
IS	Identical Suppress	SE
LC	Leading Characters	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
SF	Spacing Factor	SE
SG	Sign Position	SE
TC	Trailing Characters	SE
UC	Underlining Character	SE
ZP	Zero Printing	SE

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

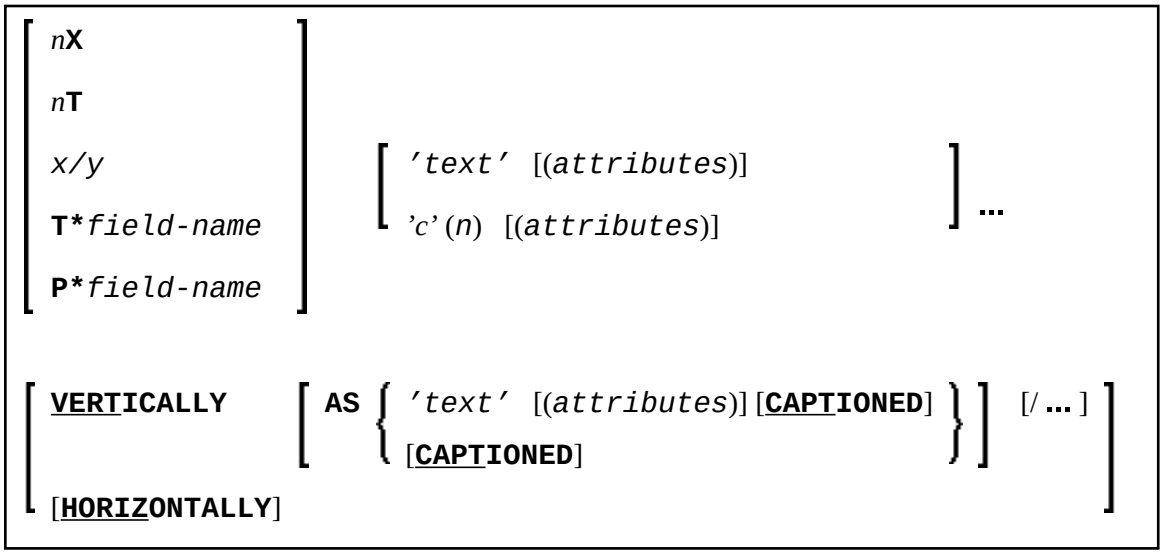
See also the following topics (in the *Programming Guide*):

- *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
*                                                          /*    ---------
DISPLAY NOHDR           'Text'          '='   VARI         /*    Text 1234
DISPLAY NOHDR (PM=I)    'Text'          '='   VARI         /*    Text 4321
DISPLAY NOHDR           'Text' (PM=I)   '='   VARI (PM=I)  /*    txeT 4321
DISPLAY NOHDR           'Text' (PM=I)   '='   VARI         /*    txeT 1234
END
```

Output Format Definitions



Field Positioning Notations

***n*X**	**Column Spacing:** This notation inserts *n* spaces between columns.*n* must not be zero. Example: `DISPLAY NAME 5X SALARY` See also: ● *Example 1 - DISPLAY Statement Using nX and nT Notation* (below) ● Column Spacing - SF Parameter and *n*X Notation (in the *Programming Guide*)
***n*T**	**Tab Setting:** The *n*T notation causes positioning (tabulation) to display position *n*. Backward positioning is not permitted. In the following example, NAME is displayed beginning in position 25, and SALARY beginning in position 50: `DISPLAY 25T NAME 50T SALARY` See also: ● *Example 1 - DISPLAY Statement Using nX and 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 is not permitted.
Tfield-name***	**Field Related Positioning:** The T* notation is used to position to a specific print position of a field used in a previous DISPLAY statement. Backward positioning is not permitted.
Pfield-name***	**Field and Line Related Positioning:** The P* notation 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 display mode. Backward positioning is not permitted. See also: ● *Example 3 - DISPLAY Statement Using P* Notation* (below) ● *Tab Notation P*field* (in the *Programming Guide*)

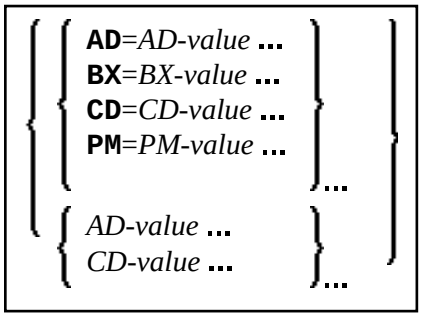
Override Column Heading Assignment

'text' *'/'*	**Text Assignment:** If placed immediately before a field, the text enclosed by single quotes overrides the column heading. The slash character `'/'` before a field causes the header for the field to be suppressed. `DISPLAY 'EMPLOYEE' NAME 'MARITAL/STATUS' MAR-STAT` If multiple `'text'` elements are specified before a field name, the *last* `'text'` element will be used as the column header and the other text elements will be placed before the value of the field within the column. See also: • *Define Your Own Column Headers* (in the *Programming Guide*) • *Text Notation, Defining a Text to Be Used with a Statement* (in the *Programming Guide*) • *Example 4 - DISPLAY Statement Using 'text', 'c(n)' and Attribute Notation* (below)

'c'(n)	**Character Repetition:**
	
	The character enclosed by single quotes is displayed *n* times immediately before the field value. For example:
	
	`DISPLAY '*' (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*)
	• *Example 4 - DISPLAY Statement Using 'text', 'c(n)' and Attribute Notation* (below)

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.

Vertical/Horizontal Display

The `VERT` clause may be used to cause multiple field values to be positioned underneath one another in the same column. In vertical mode, a new column may be initiated by specifying the keyword `VERT` or `HORIZ`.

The column heading in vertical mode is controlled using the entry or entries specified with the AS clause as described below.

VERTICALLY	Vertical column orientation. No column heading is produced if the AS clause is omitted. `DISPLAY VERT NAME SALARY` For an example, see *DISPLAY VERT without AS Clause* (in the *Programming Guide*).
AS *'text'*	Vertical column orientation. If `AS 'text'` is specified, the text enclosed by single quotes is used as the column heading. For an example, see *DISPLAY VERT AS 'text'* (in the *Programming Guide*). The slash character / in the character string of `'text'` will cause multiple lines of column headings. `DISPLAY VERT AS 'LAST/NAME' NAME`
AS *'text'* CAPTIONED	Vertical column orientation. If `AS 'text' CAPTIONED` is specified, `'text'` is used as the column heading and the standard heading text or field name is inserted immediately before the field value in each detail display line. `DISPLAY VERT AS 'PERSONS/SELECTED' CAPTIONED NAME FIRST-NAME` For an example, see *DISPLAY VERT AS 'text' CAPTIONED* (in the *Programming Guide*).
AS CAPTIONED	Vertical column orientation. If `AS CAPTIONED` is specified, the standard heading text for the field (either heading text or the field name) will be used as the column heading. `DISPLAY VERT AS CAPTIONED NAME FIRST-NAME`
HORIZONTALLY	Horizontal column orientation. This is the default display mode.

Vertical and horizontal column orientation may be intermixed by using the respective keyword.

To suspend vertical display for a single output element, you may place a dash (-) in front of the element. For example:

```
DISPLAY VERT NAME - FIRST-NAME SALARY
```

In the above example, `FIRST-NAME` will be output horizontally next to `NAME`, while `SALARY` will be output vertically again, i.e. below `NAME`.

The standard display mode is horizontal. A column is constructed for each field to be displayed.

Column headings are obtained and used by Natural according to the following priority:

1. heading `'text'` supplied in the `DISPLAY` statement;
2. the default heading defined in the DDM (database fields), or the name of a user-defined variable;
3. the field name as defined in the DDM (if no heading text was defined for the database field).

For group names, a group heading is produced for the entire group. When specifying a group, only the heading for the entire group may be overridden by a user-specified heading.

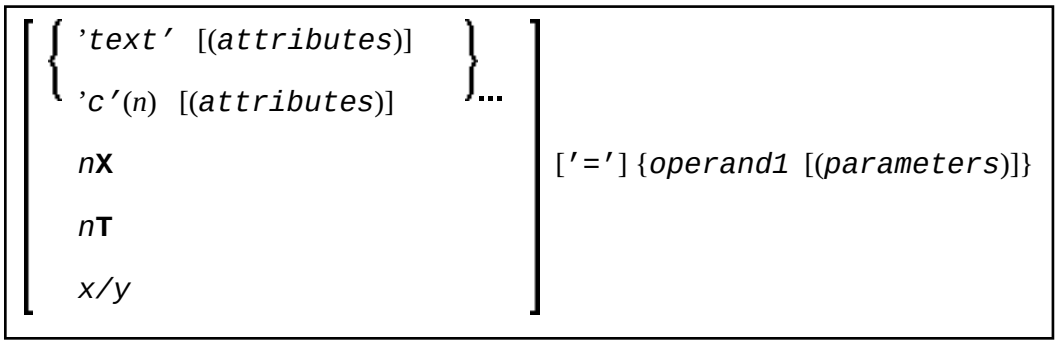
The maximum number of column header lines is 15.

Line size overflow is not permitted for output resulting from a `DISPLAY` statement. If a line overflow occurs, an error message is issued.

For more information about vertical/horizontal display usage, see:

- *Example 5 - DISPLAY Statement Using Horizontal Display*
- *Example 6 - DISPLAY Statement Using Vertical and Horizontal Display*
- *DISPLAY VERT AS CAPTIONED and HORIZ* (in the *Programming Guide*)

Output Element



Operand Definition Table:

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

Syntax Element Description

***n*X**	**Column Spacing:** This is the same as under *Output Format Definitions* (see above).
***n*T**	**Tab Setting:** This is the same as under *Output Format Definitions* (see above).
x/y	**x/y Positioning:** This is the same as under *Output Format Definitions* (see above).
'text'	**Text Assignment:** This is the same as under *Output Format Definitions* (see above).
'c'(n)	**Character Repetition:** This is the same as under *Output Format Definitions* (see above).
'text' '=' ***'c' (n) '='***	If *'text' '='* is placed immediately before the field, *text* is output immediately before the field value. This applies analogously with *'c' (n) '='*. `DISPLAY '*****' '=' NAME`
attributes	**Output Attributes:** This is the same as under *Output Format Definitions* (see above).
operand1	The field to be displayed. **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 must not be split between two statement lines. See also: • *List of Parameters* • *Example of Parameter Usage at Statement and Element (Field) Level*

Defaults Applicable for a DISPLAY Statement

The following defaults are applicable for a DISPLAY statement:

- **Report Width**
  The width of the report defaults to the value set when Natural is installed. This default value is normally 132 in batch mode or the line length of the terminal in TP mode. It may be overridden with the session parameter LS. In TP mode, line size (LS) and page size (PS) parameters are set by Natural based on the physical characteristics of the terminal type in use.

- **Terminal Screen Output**
  When the DISPLAY output is displayed on a terminal (emulation) screen, the output begins in physical Column 2 (because Column 1 must be reserved for possible use as an attribute position on a 3270-type terminal).

- **Printout on Paper**
  When the DISPLAY output is printed on paper, the printout begins in the leftmost column (Column 1).

- **Spacing Factor**
  The default spacing factor between elements is one position. There is a minimum of one space between columns (reserved for terminal attributes). This default may be overridden with the session parameter SF.

- **Field Output**
  The length of the field or the field heading, whichever is greater, determines the column width for the report (unless the HW parameter is used).

  - If the field is longer than the heading, the heading will be centered over the column unless the HC=L or HC=R parameter is used to produce a left-justified or right-justified heading.
  - If the heading is longer than the field, the field will be left-justified under the heading.
  - The values contained in the field are left-justified for alphanumeric fields and right-justified for numeric fields.
  - Numeric fields may be displayed left-justified by specifying AD=L.
  - Alphanumeric fields may be displayed right-justified by specifying AD=R.
  - In a vertical display, the longest data value or heading among all fields determines the column width (unless the HW parameter is used).

- **Sign**
  One extra high-order print position is reserved for a sign when printing a numeric field. The session parameter SG may be used to suppress the sign position.

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

Examples

- Example 1 - DISPLAY Statement Using nX and nT Notation
- Example 2 - DISPLAY Statement Using GIVE SYSTEM FUNCTIONS Clause
- Example 3 - DISPLAY Statement Using P* Notation
- Example 4 - DISPLAY Statement Using ’text ’, ’c(n)’ and Attribute Notation
- Example 5 - DISPLAY Statement Using Horizontal Display
- Example 6 - DISPLAY Statement Using Vertical and Horizontal Display
- Example 7 - DISPLAY Statement Using Parameters on Statement/Element Level
- Example 8 - Report Specification with Output File Defined to Natural as PC

Example 1 - DISPLAY Statement Using *n*X and *n*T Notation

```
** Example 'DISEX1': DISPLAY (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
  DISPLAY NOTITLE 5X NAME 50T JOB-TITLE
END-READ
*
END
```

Output of Program DISEX1:

```
        NAME                                        CURRENT
                                                   POSITION
--------------------                    -------------------------

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

Example 2 - DISPLAY Statement Using GIVE SYSTEM FUNCTIONS Clause

```
** Example 'DISEX2': DISPLAY (with GIVE SYSTEM FUNCTIONS)
************************************************************************
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 PERSONNEL-ID
  2 NAME
  2 FIRST-NAME
  2 SALARY    (1)
  2 CURR-CODE (1)
END-DEFINE
```

```
*
LIMIT 15
FORMAT PS=15
*
READ EMPLOY-VIEW
  DISPLAY GIVE SYSTEM FUNCTIONS
          PERSONNEL-ID NAME FIRST-NAME SALARY (1) CURR-CODE (1)
  AT END OF PAGE
    WRITE /    'SALARY STATISTICS:'
          / 7X 'MAXIMUM:' MAX(SALARY(1))  CURR-CODE (1)
          / 7X 'MINIMUM:' MIN(SALARY(1))  CURR-CODE (1)
          / 7X 'AVERAGE:' AVER(SALARY(1)) CURR-CODE (1)
  END-ENDPAGE
END-READ
*
END
```

Output of Program DISEX2:

```
Page      1                                                  05-01-12  09:47:48

PERSONNEL         NAME              FIRST-NAME        ANNUAL   CURRENCY
   ID                                                 SALARY     CODE
--------- -------------------- -------------------- ---------- --------

50005500  BLOND                ALEXANDRE              172000 FRA
50005300  MAIZIERE             ELISABETH              166900 FRA
50004900  CAOUDAL              ALBERT                 167350 FRA
50004600  VERDIE                BERNARD                170100 FRA
50004200  VAUZELLE             BERNARD                159790 FRA
50004100  CHAPUIS              ROBERT                 169900 FRA
50003800  JOUSSELIN            DANIEL                 171990 FRA
50006900  BAILLET              PATRICK                188000 FRA
50007600  MARX                 JEAN-MARIE             365700 FRA

SALARY STATISTICS:
       MAXIMUM:     365700 FRA
       MINIMUM:     159790 FRA
       AVERAGE:     192414 FRA
```

Example 3 - DISPLAY Statement Using P* Notation

```
** Example 'DISEX3': DISPLAY (with P* notation)
************************************************************************
DEFINE DATA LOCAL
1 EMPL-VIEW VIEW OF EMPLOYEES
  2 NAME
  2 SALARY (1)
  2 BIRTH
  2 CITY
END-DEFINE
*
LIMIT 2
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 OF CITY
    DISPLAY P*SALARY (1) AVER(SALARY (1))
```

```
    SKIP 1
  END-BREAK
END-READ
END
```

Output of Program DISEX3:

```
        NAME                 CITY           BIRTH
                                            SALARY
-------------------- -------------------- ----------

WILCOX               NASHVILLE            1970-01-01
                                               38000

MORRISON             NASHVILLE            1949-07-10
                                               36000


                                               37000
```

Example 4 - DISPLAY Statement Using *'text '*, 'c(*n*)' and Attribute Notation

```
** Example 'DISEX4': DISPLAY (with 'c(n)' notation and attribute)
************************************************************************
DEFINE DATA LOCAL
1 EMPL-VIEW VIEW OF EMPLOYEES
  2 DEPT
  2 LEAVE-DUE
  2 NAME
END-DEFINE
*
LIMIT 4
READ EMPL-VIEW BY DEPT FROM 'T'
  IF LEAVE-DUE GT 40
   DISPLAY NOTITLE
           'EMPLOYEE' NAME                /* OVERRIDE STANDARD HEADER
           'LEAVE ACCUMULATED' LEAVE-DUE  /* OVERRIDE STANDARD HEADER
           '*'(10)(I)                     /* DISPLAY 10 '*' INTENSIFIED
  ELSE
    DISPLAY NAME LEAVE-DUE
  END-IF
END-READ
*
END
```

Output of Program DISEX4:

```
      EMPLOYEE       LEAVE ACCUMULATED
-------------------- -----------------

LAVENDA               33
BOYER                 33
CORREARD              45               **********
BOUVIER               19
```

Example 5 - DISPLAY Statement Using Horizontal Display

```
** Example 'DISEX5': DISPLAY (horizontal display)
************************************************************************
DEFINE DATA LOCAL
1 EMPL-VIEW VIEW OF EMPLOYEES
  2 NAME
  2 JOB-TITLE
  2 SALARY    (1:2)
  2 CURR-CODE (1:2)
END-DEFINE
*
LIMIT 4
READ EMPL-VIEW BY NAME
 DISPLAY NOTITLE NAME JOB-TITLE SALARY (1:2) CURR-CODE (1:2)
  SKIP 1
END-READ
*
END
```

Output of Program DISEX5:

```
        NAME                  CURRENT            ANNUAL   CURRENCY
                             POSITION            SALARY     CODE
------------------- ------------------------ ---------- --------

ABELLAN             MAQUINISTA                  1450000 PTA
                                                1392000 PTA

ACHIESON            DATA BASE ADMINISTRATOR       11300 UKL
                                                  10500 UKL

ADAM                CHEF DE SERVICE              159980 FRA
                                                      0

ADKINSON            PROGRAMMER                    34500 USD
                                                  31700 USD
```

Example 6 - DISPLAY Statement Using Vertical and Horizontal Display

```
** Example 'DISEX6': DISPLAY (vertical and horizontal display)
************************************************************************
DEFINE DATA LOCAL
1 EMPL-VIEW VIEW OF EMPLOYEES
  2 NAME
  2 CITY
  2 JOB-TITLE
  2 SALARY    (1:2)
  2 CURR-CODE (1:2)
END-DEFINE
*
LIMIT 1
READ EMPL-VIEW BY NAME
 DISPLAY NOTITLE VERT AS CAPTIONED
          NAME CITY 'POSITION' JOB-TITLE
          HORIZ 'SALARY' SALARY (1:2) 'CURRENCY' CURR-CODE (1:2)
  /*
  SKIP 1
END-READ
END
```

Output of Program DISEX6:

```
          NAME              SALARY   CURRENCY
          CITY
        POSITION
------------------------ ---------- --------

ABELLAN                     1450000 PTA
MADRID                      1392000 PTA
MAQUINISTA
```

Example 7 - DISPLAY Statement Using Parameters on Statement/Element Level

```
** Example 'DISEX7': DISPLAY (with parameters for statement/element)
************************************************************************
DEFINE DATA LOCAL
1 EMPL-VIEW VIEW OF EMPLOYEES
  2 NAME
  2 PERSONNEL-ID
  2 TELEPHONE
    3 AREA-CODE
    3 PHONE
END-DEFINE
*
LIMIT 3
READ EMPL-VIEW BY NAME
 DISPLAY NOTITLE (AL=16 GC=+ NL=8 SF=3 UC==)
          PERSONNEL-ID NAME TELEPHONE (LC=< TC=>)
END-READ
END
```

Output of Program DISEX7:

```
   PERSONNEL             NAME         +++++++++++++++TELEPHONE+++++++++++++++
       ID
                                             AREA              TELEPHONE
                                             CODE
================   ================   ==================   ==================

60008339           ABELLAN            <1     >             <4356726        >
30000231           ACHIESON           <0332  >             <523341         >
50005800           ADAM               <1033  >             <44864858       >
```

Example 8 - 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
```