

Vertical Displays

This chapter describes how you can combine the features of the statements `DISPLAY` and `WRITE` to produce vertical displays of field values.

The following topics are covered:

- Creating Vertical Displays
 - Combining `DISPLAY` and `WRITE`
 - Tab Notation - `T*field`
 - Positioning Notation `x/y`
 - `DISPLAY VERT` Statement
 - Further Example of `DISPLAY VERT` with `WRITE` Statement
-

Creating Vertical Displays

There are two ways of creating vertical displays:

- You can use a combination of the statements `DISPLAY` and `WRITE`.
- You can use the `VERT` option of the `DISPLAY` statement.

Combining `DISPLAY` and `WRITE`

As described in *Statements `DISPLAY` and `WRITE`*, the `DISPLAY` statement normally presents the data in columns with default headers, while the `WRITE` statement presents data horizontally without headers.

You can combine the features of the two statements to produce vertical displays of field values.

The `DISPLAY` statement produces the values of different fields for the same record across the page with a column for each field. The field values for each record are displayed below the values for the previous record.

By using a `WRITE` statement after a `DISPLAY` statement, you can insert text and/or field values specified in the `WRITE` statement between records displayed via the `DISPLAY` statement.

The following program illustrates the combination of `DISPLAY` and `WRITE`:

```
** Example 'WRITEX04': WRITE (in combination with DISPLAY)
*****
DEFINE DATA LOCAL
1 VIEWEMP VIEW OF EMPLOYEES
  2 NAME
  2 JOB-TITLE
  2 CITY
  2 DEPT
END-DEFINE
```

```

*
READ (3) VIEWEMP BY CITY STARTING FROM 'SAN FRANCISCO'
  DISPLAY NAME JOB-TITLE
  WRITE 22T 'DEPT:' DEPT
  SKIP 1
END-READ
END

```

Output of Program WRITEX04:

```

Page          1                                04-11-11  14:15:55

      NAME                                CURRENT
      POSITION
-----
KOLENCE                                MANAGER
                                         DEPT: TECH05

GOSDEN                                  ANALYST
                                         DEPT: TECH10

WALLACE                                SALES PERSON
                                         DEPT: SALE20

```

Tab Notation - T*field

In the previous example, the position of the field DEPT is determined by the tab notation *nT* (in this case 20T, which means that the display begins in column 20 on the screen).

Field values specified in a WRITE statement can be lined up automatically with field values specified in the first DISPLAY statement of the program by using the tab notation T**field* (where *field* is the name of the field to which the field is to be aligned).

In the following program, the output produced by the WRITE statement is aligned to the field JOB-TITLE by using the notation T*JOB-TITLE:

```

** Example 'WRITEX05': WRITE (in combination with DISPLAY)
*****
DEFINE DATA LOCAL
1 VIEWEMP VIEW OF EMPLOYEES
  2 NAME
  2 JOB-TITLE
  2 DEPT
  2 CITY
END-DEFINE
*
READ (3) VIEWEMP BY CITY STARTING FROM 'SAN FRANCISCO'
  DISPLAY NAME JOB-TITLE
  WRITE T*JOB-TITLE 'DEPT:' DEPT
  SKIP 1
END-READ
END

```

Output of Program WRITEX05:

Page 1

04-11-11 14:15:55

NAME	CURRENT POSITION
-----	-----
KOLENCE	MANAGER DEPT: TECH05
GOSDEN	ANALYST DEPT: TECH10
WALLACE	SALES PERSON DEPT: SALE20

Positioning Notation x/y

When you use the DISPLAY and WRITE statements in sequence and multiple lines are to be produced by the WRITE statement, you can use the notation x/y (number-slash-number) to determine in which row/column something is to be displayed. The positioning notation causes the next element in the DISPLAY or WRITE statement to be placed x lines below the last output, beginning in column y of the output.

The following program illustrates the use of this notation:

```

** Example 'WRITEX06': WRITE (with n/n)
*****
DEFINE DATA LOCAL
1 VIEWEMP VIEW OF EMPLOYEES
  2 NAME
  2 FIRST-NAME
  2 MIDDLE-I
  2 ADDRESS-LINE (1:1)
  2 CITY
  2 ZIP
END-DEFINE
*
READ (3) VIEWEMP BY CITY STARTING FROM 'NEW YORK'
  DISPLAY 'NAME AND ADDRESS' NAME
  WRITE 1/5 FIRST-NAME
      1/30 MIDDLE-I
      2/5 ADDRESS-LINE (1:1)
      3/5 CITY
      3/30 ZIP /
END-READ
END

```

Output of Program WRITEX06:

Page 1

04-11-11 14:15:55

NAME AND ADDRESS	
-----	-----
RUBIN	
SYLVIA	L
2003 SARAZEN PLACE	
NEW YORK	10036

```

WALLACE
  MARY                P
  12248 LAUREL GLADE C
  NEW YORK            10036

KELLOGG
  HENRIETTA          S
  1001 JEFF RYAN DR.
  NEWARK             19711

```

DISPLAY VERT Statement

The standard display mode in Natural is horizontal.

With the VERT clause option of the DISPLAY statement, you can override the standard display and produce a vertical field display.

The HORIZ clause option, which can be used in the same DISPLAY statement, re-activates the standard horizontal display mode.

Column headings in vertical mode are controlled with various forms of the AS clause. The following example programs illustrate the use of the DISPLAY VERT statement:

- DISPLAY VERT without AS Clause
- DISPLAY VERT AS CAPTIONED and HORIZ
- DISPLAY VERT AS 'text'
- DISPLAY VERT AS 'text' CAPTIONED
- Tab Notation P*field

DISPLAY VERT without AS Clause

The following program has no AS clause, which means that no column headings are output.

```

** Example 'DISPLX09': DISPLAY (without column title)
*****
DEFINE DATA LOCAL
1 VIEWEMP VIEW OF EMPLOYEES
  2 NAME
  2 FIRST-NAME
  2 CITY
END-DEFINE
*
READ (3) VIEWEMP BY CITY STARTING FROM 'NEW YORK'
  DISPLAY VERT NAME FIRST-NAME / CITY
  SKIP 2
END-READ
END

```

Output of Program DISPLX09:

Note that all field values are displayed vertically underneath one another.

Page 1

04-11-11 14:15:54

RUBIN
SYLVIA

NEW YORK

WALLACE
MARY

NEW YORK

KELLOGG
HENRIETTA

NEWARK

DISPLAY VERT AS CAPTIONED and HORIZ

The following program contains a VERT and a HORIZ clause, which causes some column values to be output vertically and others horizontally; moreover AS CAPTIONED causes the default column headers to be displayed.

```

** Example 'DISPLX10': DISPLAY (with VERT as CAPTIONED and HORIZ clause)
*****
DEFINE DATA LOCAL
1 VIEWEMP VIEW OF EMPLOYEES
  2 NAME
  2 FIRST-NAME
  2 CITY
  2 JOB-TITLE
  2 SALARY (1:1)
END-DEFINE
*
READ (3) VIEWEMP BY CITY STARTING FROM 'NEW YORK'
  DISPLAY VERT AS CAPTIONED NAME FIRST-NAME
  HORIZ JOB-TITLE SALARY (1:1)
  SKIP 1
END-READ
END
    
```

Output of Program DISPLX10:

Page 1

04-11-11 14:15:54

NAME FIRST-NAME	CURRENT POSITION	ANNUAL SALARY
RUBIN SYLVIA	SECRETARY	17000
WALLACE	ANALYST	38000

MARY

KELLOGG
HENRIETTA

DIRECTOR

52000

DISPLAY VERT AS 'text'

The following program contains an AS 'text' clause, which displays the specified 'text' as column header.

Note:

A slash (/) within the text element in a DISPLAY statement causes a line advance.

```
** Example 'DISPLX11': DISPLAY (with VERT AS 'text' clause)
*****
DEFINE DATA LOCAL
1 VIEWEMP VIEW OF EMPLOYEES
  2 NAME
  2 FIRST-NAME
  2 CITY
  2 JOB-TITLE
  2 SALARY (1:1)
END-DEFINE
*
READ (3) VIEWEMP BY CITY STARTING FROM 'NEW YORK'
  DISPLAY VERT AS 'EMPLOYEES' NAME FIRST-NAME
    HORIZ JOB-TITLE SALARY (1:1)
  SKIP 1
END-READ
END
```

Output of Program DISPLX11:

```
Page      1                                04-11-11  14:15:54
```

EMPLOYEES	CURRENT POSITION	ANNUAL SALARY
RUBIN SYLVIA	SECRETARY	17000
WALLACE MARY	ANALYST	38000
KELLOGG HENRIETTA	DIRECTOR	52000

DISPLAY VERT AS 'text' CAPTIONED

The AS 'text' CAPTIONED clause causes the specified text to be displayed as column heading, and the default column headings to be displayed immediately before the field value in each line that is output.

The following program contains an AS 'text' CAPTIONED clause.

```
** Example 'DISPLX12': DISPLAY (with VERT AS 'text' CAPTIONED clause)
*****
DEFINE DATA LOCAL
1 VIEWEMP VIEW OF EMPLOYEES
```

```

2 NAME
2 FIRST-NAME
2 CITY
2 JOB-TITLE
2 SALARY (1:1)
END-DEFINE
*
READ (3) VIEWEMP BY CITY STARTING FROM 'NEW YORK'
  DISPLAY VERT AS 'EMPLOYEES' CAPTIONED NAME FIRST-NAME
    HORIZ JOB-TITLE SALARY (1:1)
  SKIP 1
END-READ
END

```

Output of Program DISPLX12:

This clause causes the default column headers (NAME and FIRST-NAME) to be placed before the field values:

Page 1 04-11-11 14:15:54

EMPLOYEES	CURRENT POSITION	ANNUAL SALARY
NAME RUBIN FIRST-NAME SYLVIA	SECRETARY	17000
NAME WALLACE FIRST-NAME MARY	ANALYST	38000
NAME KELLOGG FIRST-NAME HENRIETTA	DIRECTOR	52000

Tab Notation P*field

If you use a combination of DISPLAY VERT statement and subsequent WRITE statement, you can use the tab notation P*field-name in the WRITE statement to align the position of a field to the column and line position of a particular field specified in the DISPLAY VERT statement.

In the following program, the fields SALARY and BONUS are displayed in the same column, SALARY in every first line, BONUS in every second line. The text ***SALARY PLUS BONUS*** is aligned to SALARY, which means that it is displayed in the same column as SALARY and in the first line, whereas the text (IN US DOLLARS) is aligned to BONUS and therefore displayed in the same column as BONUS and in the second line.

```

** Example 'WRITEX07': WRITE (with P*field)
*****
DEFINE DATA LOCAL
1 VIEWEMP VIEW OF EMPLOYEES
  2 CITY
  2 NAME
  2 JOB-TITLE
  2 SALARY (1:1)
  2 BONUS (1:1,1:1)
END-DEFINE
*
READ (3) VIEWEMP BY CITY STARTING FROM 'LOS ANGELES'
  DISPLAY NAME JOB-TITLE

```

```

      VERT AS 'INCOME' SALARY (1) BONUS (1,1)
WRITE P*SALARY '***SALARY PLUS BONUS***'
      P*BONUS '(IN US DOLLARS)'
SKIP 1
END-READ
END

```

Output of Program WRITEX07:

```

Page      1                                04-11-11  14:15:55

      NAME                                CURRENT          INCOME
      POSITION
-----
SMITH                                0
                                0
                                ***SALARY PLUS BONUS***
                                (IN US DOLLARS)

POORE JR                            SECRETARY          25000
                                0
                                ***SALARY PLUS BONUS***
                                (IN US DOLLARS)

PREPARATA                            MANAGER           46000
                                9000
                                ***SALARY PLUS BONUS***
                                (IN US DOLLARS)

```

Further Example of DISPLAY VERT with WRITE Statement

See the following example program:

- *WRITEX10* - WRITE (with *nT*, *T*field* and *P*field*)