

Statements DISPLAY and WRITE

This chapter describes how to use the statements DISPLAY and WRITE to output data and control the format in which information is output.

The following topics are covered:

- DISPLAY Statement
 - WRITE Statement
 - Example of DISPLAY Statement
 - Example of WRITE Statement
 - Column Spacing - SF Parameter and nX Notation
 - Tab Setting - nT Notation
 - Line Advance - Slash Notation
 - Further Examples of DISPLAY and WRITE Statements
-

DISPLAY Statement

The DISPLAY statement produces output in column format; that is, the values for one field are output in a column underneath one another. If multiple fields are output, that is, if multiple columns are produced, these columns are output next to one another horizontally.

The order in which fields are displayed is determined by the sequence in which you specify the field names in the DISPLAY statement.

The DISPLAY statement in the following program displays for each person first the personnel number, then the name and then the job title:

```
** Example 'DISPLX01': DISPLAY
*****
DEFINE DATA LOCAL
1 VIEWEMP VIEW OF EMPLOYEES
  2 PERSONNEL-ID
  2 NAME
  2 BIRTH
  2 JOB-TITLE
END-DEFINE
*
READ (3) VIEWEMP BY BIRTH
  DISPLAY PERSONNEL-ID NAME JOB-TITLE
END-READ
END
```

Output of Program DISPLX01:

Page 1

04-11-11 14:15:54

PERSONNEL ID	NAME	CURRENT POSITION
30020013	GARRET	TYPIST
30016112	TAILOR	WAREHOUSEMAN
20017600	PIETSCH	SECRETARY

To change the order of the columns that appear in the output report, simply reorder the field names in the DISPLAY statement. For example, if you prefer to list employee names first, then job titles followed by personnel numbers, the appropriate DISPLAY statement would be:

```

** Example 'DISPLX02': DISPLAY
*****
DEFINE DATA LOCAL
1 VIEWEMP VIEW OF EMPLOYEES
  2 PERSONNEL-ID
  2 NAME
  2 BIRTH
  2 JOB-TITLE
END-DEFINE
*
READ (3) VIEWEMP BY BIRTH
  DISPLAY NAME JOB-TITLE PERSONNEL-ID
END-READ
END
    
```

Output of Program DISPLX02:

Page 1

04-11-11 14:15:54

NAME	CURRENT POSITION	PERSONNEL ID
GARRET	TYPIST	30020013
TAILOR	WAREHOUSEMAN	30016112
PIETSCH	SECRETARY	20017600

A header is output above each column. Various ways to influence this header are described in the document *Column Headers*.

WRITE Statement

The WRITE statement is used to produce output in free format (that is, not in columns). In contrast to the DISPLAY statement, the following applies to the WRITE statement:

- If necessary, it automatically creates a line advance; that is, a field or text element that does not fit onto the current output line, is automatically output in the next line.
- It does not produce any headers.
- The values of a multiple-value field are output next to one another horizontally, and not underneath one another.

The two example programs shown below illustrate the basic differences between the DISPLAY statement and the WRITE statement.

You can also use the two statements in combination with one another, as described later in the document *Vertical Displays, Combining DISPLAY and WRITE* .

Example of DISPLAY Statement

```

** Example 'DISPLX03': DISPLAY
*****
DEFINE DATA LOCAL
1 VIEWEMP VIEW OF EMPLOYEES
  2 NAME
  2 FIRST-NAME
  2 SALARY (1:3)
END-DEFINE
*
READ (2) VIEWEMP BY NAME STARTING FROM 'JONES'
  DISPLAY NAME FIRST-NAME SALARY (1:3)
END-READ
END
    
```

Output of Program DISPLX03:

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

NAME	FIRST-NAME	ANNUAL SALARY
JONES	VIRGINIA	46000
		42300
		39300
JONES	MARSHA	50000
		46000
		42700

Example of WRITE Statement

```

** Example 'WRITEX01': WRITE
*****
DEFINE DATA LOCAL
1 VIEWEMP VIEW OF EMPLOYEES
  2 NAME
  2 FIRST-NAME
  2 SALARY (1:3)
END-DEFINE
*
READ (2) VIEWEMP BY NAME STARTING FROM 'JONES'
  WRITE NAME FIRST-NAME SALARY (1:3)
END-READ
END
    
```

Output of Program WRITEX01:

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

JONES	VIRGINIA	46000	42300	39300
JONES	MARSHA	50000	46000	42700

Column Spacing - SF Parameter and *nX* Notation

By default, the columns output with a DISPLAY statement are separated from one another by *one* space.

With the session parameter SF, you can specify the default number of spaces to be inserted between columns output with a DISPLAY statement. You can set the number of spaces to any value from 1 to 30.

The parameter can be specified with a FORMAT statement to apply to the whole report, or with a DISPLAY statement at statement level, but not at element level.

With the *nX* notation in the DISPLAY statement, you can specify the number of spaces (*n*) to be inserted between two columns. An *nX* notation overrides the specification made with the SF parameter.

```
** Example 'DISPLX04': DISPLAY (with nX)
*****
DEFINE DATA LOCAL
1 VIEWEMP VIEW OF EMPLOYEES
  2 PERSONNEL-ID
  2 NAME
  2 BIRTH
  2 JOB-TITLE
END-DEFINE
*
FORMAT SF=3
READ (3) VIEWEMP BY BIRTH
  DISPLAY PERSONNEL-ID NAME 5X JOB-TITLE
END-READ
END
```

Output of Program DISPLX04:

The above example program produces the following output, where the first two columns are separated by 3 spaces due to the SF parameter in the FORMAT statement, while the second and third columns are separated by 5 spaces due to the notation 5X in the DISPLAY statement:

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

PERSONNEL          NAME                      CURRENT
  ID              -----                POSITION
-----
30020013    GARRET                      TYPIST
30016112    TAILOR                        WAREHOUSEMAN
20017600    PIETSCH                          SECRETARY
```

The *nX* notation is also available with the WRITE statement to insert spaces between individual output elements:

```
WRITE PERSONNEL-ID 5X NAME 3X JOB-TITLE
```

With the above statement, 5 spaces will be inserted between the fields PERSONNEL-ID and NAME, and 3 spaces between NAME and JOB-TITLE.

Tab Setting - nT Notation

With the *nT* notation, which is available with the DISPLAY and the WRITE statement, you can specify the position where an output element is to be output.

```
** Example 'DISPLX05': DISPLAY (with nT)
*****
DEFINE DATA LOCAL
1 VIEWEMP VIEW OF EMPLOYEES
  2 NAME
  2 FIRST-NAME
END-DEFINE
*
READ (3) VIEWEMP BY NAME STARTING FROM 'JONES'
  DISPLAY 5T NAME 30T FIRST-NAME
END-READ
END
```

Output of Program DISPLX05:

The above program produces the following output, where the field NAME is output starting in the 5th position (counted from the left margin of the page), and the field FIRST-NAME starting in the 30th position:

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

          NAME                                FIRST-NAME
-----
JONES                                VIRGINIA
JONES                                MARSHA
JONES                                ROBERT
```

Line Advance - Slash Notation

With a slash (/) in a DISPLAY or WRITE statement, you cause a line advance.

- In a DISPLAY statement, a slash causes a line advance *between fields* and *within text*.
- In a WRITE statement, a slash causes a line advance only when placed *between fields*; within text, it is treated like an ordinary text character.

When placed between fields, the slash must have a blank on either side.

For multiple line advances, you specify multiple slashes.

Example 1 - Line Advance in DISPLAY Statement:

```
** Example 'DISPLX06': DISPLAY (with slash '/')
*****
DEFINE DATA LOCAL
1 VIEWEMP VIEW OF EMPLOYEES
  2 NAME
  2 FIRST-NAME
  2 DEPARTMENT
END-DEFINE
*
```

```

READ (3) VIEWEMP BY NAME STARTING FROM 'JONES'
  DISPLAY NAME / FIRST-NAME 'DEPART-/MENT' DEPARTMENT
END-READ
END
    
```

Output of Program DISPLX06:

The above DISPLAY statement produces a line advance after each value of the field NAME and within the text DEPART-MENT:

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

NAME FIRST-NAME	DEPART- MENT
JONES	SALE
VIRGINIA	
JONES	MGMT
MARSHA	
JONES	TECH
ROBERT	

Example 2 - Line Advance in WRITE Statement:

```

** Example 'WRITEX02': WRITE (with line advance)
*****
DEFINE DATA LOCAL
1 VIEWEMP VIEW OF EMPLOYEES
  2 NAME
  2 FIRST-NAME
  2 DEPARTMENT
END-DEFINE
*
READ (3) VIEWEMP BY NAME STARTING FROM 'JONES'
  WRITE NAME / FIRST-NAME 'DEPART-/MENT' DEPARTMENT //
END-READ
END
    
```

Output of Program WRITEX02:

The above WRITE statement produces a line advance after each value of the field NAME, and a double line advance after each value of the field DEPARTMENT, but none within the text DEPART-/MENT:

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

JONES	
VIRGINIA	DEPART-/MENT SALE
JONES	
MARSHA	DEPART-/MENT MGMT
JONES	
ROBERT	DEPART-/MENT TECH

Example 3 - Line Advance in DISPLAY and WRITE:

```

** Example 'DISPLX21': DISPLAY (usage of slash '/' in DISPLAY and WRITE)
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 CITY
  2 NAME
  2 FIRST-NAME
  2 ADDRESS-LINE (1)
END-DEFINE
*
WRITE TITLE LEFT JUSTIFIED UNDERLINED
  *TIME
  5X 'PEOPLE LIVING IN SALT LAKE CITY'
  21X 'PAGE:' *PAGE-NUMBER /
  15X 'AS OF' *DATE //
*
WRITE TRAILER UNDERLINED 'REGISTER OF' / 'SALT LAKE CITY'
*
READ (2) EMPLOY-VIEW WITH CITY = 'SALT LAKE CITY'
  DISPLAY NAME /
  FIRST-NAME
  'HOME/CITY' CITY
  'STREET/OR BOX NO.' ADDRESS-LINE (1)

  SKIP 1
END-READ
END

```

Output of Program DISPLX21:

```

14:15:54.6      PEOPLE LIVING IN SALT LAKE CITY      PAGE:      1
                AS OF 11/11/2004

```

```

-----
          NAME                HOME                STREET
          FIRST-NAME          CITY                OR BOX NO.
-----
ANDERSON                SALT LAKE CITY        3701 S. GEORGE MASON
JENNY
SAMUELSON                SALT LAKE CITY        7610 W. 86TH STREET
MARTIN

                REGISTER OF
                SALT LAKE CITY
-----

```

Further Examples of DISPLAY and WRITE Statements

See the following example programs:

- *DISPLX13 - DISPLAY (compare with WRITEX08 using WRITE)*
- *WRITEX08 - WRITE (compare with DISPLX13 using DISPLAY)*

- *DISPLX14 - DISPLAY (with AL, SF and nX)*
- *WRITEX09 - WRITE (in combination with AT END OF DATA)*