

MOVE ALL

MOVE ALL *operand1* **TO** *operand2* [**UNTIL** *operand3*]

This chapter covers the following topics:

- Function
- Syntax Description
- Example

Related Statements: ADD | COMPRESS | COMPUTE | DIVIDE | EXAMINE | MOVE | MULTIPLY | RESET | SEPARATE | SUBTRACT

Belongs to Function Group: *Arithmetic and Data Movement Operations*

Function

The **MOVE ALL** statement is used to move repeatedly the value of *operand1* to *operand2* until *operand3* is full.

Syntax Description

Operand Definition Table:

Operand	Possible Structure				Possible Formats												Referencing Permitted	Dynamic Definition		
<i>operand1</i>	C	S			A	U	N						B						yes	no
<i>operand2</i>		S	A		A	U							B						yes	yes
<i>operand3</i>	C	S								N	P	I							yes	no

Syntax Element Description:

<i>operand1</i>	<p>Source Operand:</p> <p>The source operand contains the value to be moved.</p> <p>All digits of a numeric operand including leading zeros are moved</p>
TO <i>operand2</i>	<p>Target Operand:</p> <p>The target operand is not reset prior to the execution of the MOVE ALL operation. This is of particular importance when using the UNTIL option since data previously in <i>operand2</i> is retained if not explicitly overlaid during the MOVE ALL operation.</p>
UNTIL <i>operand3</i>	<p>UNTIL Option:</p> <p>The UNTIL option is used to limit the MOVE ALL operation to a given number of positions in <i>operand2</i>. <i>Operand3</i> is used to specify the number of positions. The MOVE ALL operation is terminated when this value is reached.</p> <p>If <i>operand3</i> is greater than the length of <i>operand2</i>, the MOVE ALL operation is terminated when <i>operand2</i> is full.</p> <p>The UNTIL option may also be used to assign an initial value to a dynamic variable: if <i>operand2</i> is a dynamic variable, its length after the MOVE ALL operation will correspond to the value of <i>operand3</i>. The current length of a dynamic variable can be ascertained by using the system variable *LENGTH. For general information on dynamic variables, see <i>Usage of Dynamic Variables</i>.</p>

Example

```

** Example 'MOAEX1': MOVE ALL
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 PERSONNEL-ID
  2 FIRST-NAME
  2 NAME
  2 CITY
1 VEH-VIEW VIEW OF VEHICLES
  2 PERSONNEL-ID
  2 MAKE
END-DEFINE
*
LIMIT 4
RD. READ EMPLOY-VIEW BY NAME
  SUSPEND IDENTICAL SUPPRESS
/*
FD. FIND VEH-VIEW WITH PERSONNEL-ID = PERSONNEL-ID (RD.)
  IF NO RECORDS FOUND
    MOVE ALL '*' TO FIRST-NAME (RD.)
    MOVE ALL '*' TO CITY (RD.)
    MOVE ALL '*' TO MAKE (FD.)
  END-NOREC
/*
DISPLAY NOTITLE (ES=OFF IS=ON ZP=ON AL=15)
  NAME (RD.) FIRST-NAME (RD.)
  CITY (RD.)

```

```

                MAKE (FD.) (IS=OFF)
        /*
        END-FIND
        END-READ
        END
    
```

Output of Program MOAEX1:

NAME	FIRST-NAME	CITY	MAKE
ABELLAN	*****	*****	*****
ACHIESON	ROBERT	DERBY	FORD
ADAM	*****	*****	*****
ADKINSON	JEFF	BROOKLYN	GENERAL MOTORS