

# RESET

```
RESET [INITIAL] operand1 ...
```

This chapter covers the following topics:

- Function
- Syntax Description
- Example

For an explanation of the symbols used in the syntax diagram, see *Syntax Symbols*.

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

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

---

## Function

The RESET statement is used to reset the value of a field:

- RESET (without INITIAL) sets the content of each specified field to its default initial value depending on its format.
- RESET INITIAL sets each specified field to the initial value as defined for the field in the DEFINE DATA statement. For a field declared without INIT clause in the DEFINE DATA statement, RESET INITIAL has the same effect as RESET (without INITIAL).

### Notes:

1. A field declared with a CONSTANT clause in the DEFINE DATA statement may not be referenced in a RESET statement, since its content cannot be changed.
2. In reporting mode, the RESET statement may also be used to define a variable, provided that the program contains no DEFINE DATA LOCAL statement.

## Syntax Description

Operand Definition Table:

Operand	Possible Structure	Possible Formats	Referencing Permitted	Dynamic Definition
<i>operand1</i>	S A G M	A U N P I F B D T L C G O	yes	yes

Syntax Element Description:

<p><b>RESET</b> <i>operand1</i></p>	<p><b>Reset to Null Value:</b></p> <p>RESET (without INITIAL) sets the content of each specified field (<i>operand1</i>) to its default initial value.</p> <p>If <i>operand1</i> is a dynamic variable, it will be reset to a null value with the length the variable currently has at the time the RESET statement is executed. The current length of a dynamic variable can be ascertained by using the system variable *LENGTH.</p> <p>For general information on dynamic variables, see the section <i>Using Dynamic and Large Variables</i>.</p>
<p><b>RESET INITIAL</b> <i>operand1</i></p>	<p><b>Reset to Initial Value:</b></p> <p>RESET INITIAL sets each specified field (<i>operand1</i>) to the initial value as defined for the field in the DEFINE DATA statement.</p> <ul style="list-style-type: none"> <li>● If you specify no INIT value in the DEFINE DATA statement, a field will be initialized with a default initial value depending on its format.</li> <li>● If a dynamic variable is used, *LENGTH is set to zero if no initial value is defined.</li> <li>● If you apply RESET INITIAL to an array, it must be applied to the entire array (as defined in the DEFINE DATA statement); a RESET INITIAL of individual array occurrences is not possible.</li> <li>● RESET INITIAL of fields resulting from a redefinition is not possible either.</li> <li>● RESET INITIAL is applied to a dynamic variable.</li> <li>● RESET INITIAL cannot be applied to database fields.</li> </ul>

## Example

```

** Example 'RSTEX1': RESET (with/without INITIAL)
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 NAME
1 #BINARY (B4) INIT <1>
1 #INTEGER (I4) INIT <5>
1 #NUMERIC (N2) INIT <25>
END-DEFINE
*
LIMIT 1
READ EMPLOY-VIEW
/*
WRITE NOTITLE 'VALUES BEFORE RESET STATEMENT:'
WRITE / '=' NAME '=' #BINARY '=' #INTEGER '=' #NUMERIC
/*
RESET NAME #BINARY #INTEGER #NUMERIC

```

```

/*
WRITE /// 'VALUES AFTER RESET STATEMENT:'
WRITE / '=' NAME '=' #BINARY '=' #INTEGER '=' #NUMERIC
/*
RESET INITIAL #BINARY #INTEGER #NUMERIC
/*
WRITE /// 'VALUES AFTER RESET INITIAL STATEMENT:'
WRITE / '=' NAME '=' #BINARY '=' #INTEGER '=' #NUMERIC
/*
END-READ
END

```

### Output of Program RSTEX1:

VALUES BEFORE RESET STATEMENT:

```

NAME: ADAM                #BINARY: 00000001 #INTEGER:          5 #NUMERIC:
25

```

VALUES AFTER RESET STATEMENT:

```

NAME:                    #BINARY: 00000000 #INTEGER:          0 #NUMERIC:
0

```

VALUES AFTER RESET INITIAL STATEMENT:

```

NAME:                    #BINARY: 00000001 #INTEGER:          5 #NUMERIC:
25

```