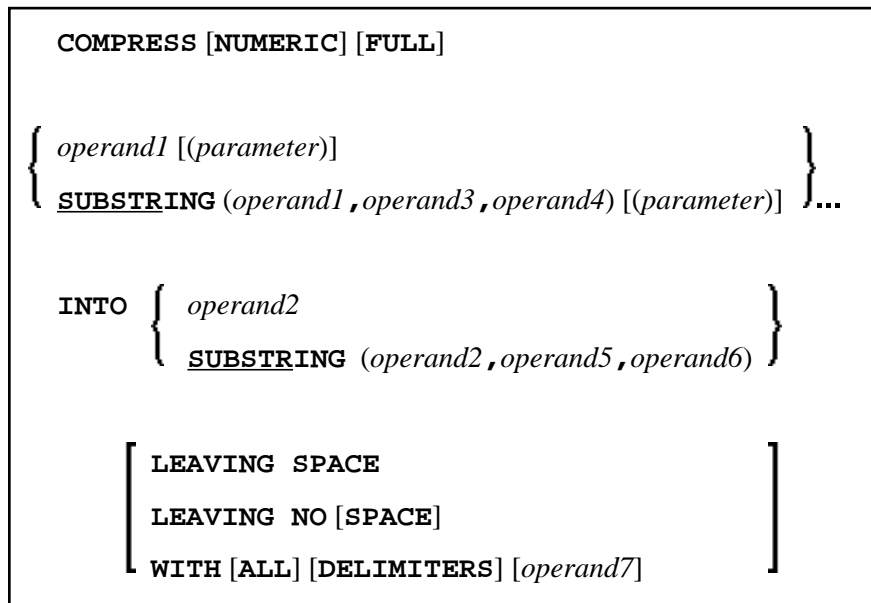


COMPRESS



This chapter covers the following topics:

- Function
- Syntax Description
- Processing
- Examples

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

Related Statements: ASSIGN | COMPUTE | EXAMINE | MOVE | MOVE ALL | SEPARATE

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

Function

The COMPRESS statement is used to transfer (combine) the contents of one or more operands into a single field.

Syntax Description

Operand Definition Table:

| Operand | Possible Structure | | | | | Possible Formats | | | | | | | | | | Referencing Permitted | Dynamic Definition | | | | |
|-----------------|--------------------|---|---|---|---|------------------|---|---|---|---|---|----|---|---|--|-----------------------|--------------------|---|---|-----|-----|
| <i>operand1</i> | C | S | A | G | N | A | U | N | P | I | F | B | D | T | | | | G | O | yes | no |
| <i>operand2</i> | | S | | | | A | U | | | | | B | | | | | | | | yes | yes |
| <i>operand3</i> | C | S | | | | | | N | P | I | | B* | | | | | | | | yes | no |
| <i>operand4</i> | C | S | | | | | | N | P | I | | B* | | | | | | | | yes | no |
| <i>operand5</i> | C | S | | | | | | N | P | I | | B* | | | | | | | | yes | no |
| <i>operand6</i> | C | S | | | | | | N | P | I | | B* | | | | | | | | yes | no |
| <i>operand7</i> | C | S | | | | A | U | | | | | B | | | | | | | | yes | no |

* Format B of *operand3*, *operand4*, *operand5* and *operand6* may be used only with a length of less than or equal to 4.

Syntax Element Description:

| Syntax Element | Description |
|----------------|--|
| NUMERIC | <p>Handling of Sign Characters:</p> <p>This option determines how sign characters and decimal characters are to be handled:</p> <p>Without NUMERIC, decimal points and signs in numeric source values are suppressed before the values are transferred. For example:</p> <pre>COMPRESS -123 1.23 INTO #TARGET WITH DELIMITER '*' Content of #TARGET is: 123*123</pre> <p>With NUMERIC, decimal points and signs in numeric source values are also transferred to the target field.</p> <p>For floating point source values, decimal points and signs are transferred, regardless of whether NUMERIC has been specified or not.</p> <p>Example 1:</p> <pre>COMPRESS NUMERIC -123 1.23 INTO #TARGET WITH DELIMITER '*' Content of #TARGET is: -123*1.23</pre> <p>Example 2:</p> <pre>COMPRESS NUMERIC 'ABC' -0056.00 -0056.10 -0056.01 INTO #TARGET WITH DELIMITER '*' Content of #TARGET is: ABC*-56*-56.1*-56.01</pre> <p>Example 3:</p> <pre>COMPRESS NUMERIC FULL 'ABC' -0056.00 -0056.10 -0056.01 INTO #TARGET WITH DELIMITER '*' Content of #TARGET is: ABC*-0056.00*-0056.10*-0056.01</pre> |

| Syntax Element | Description |
|------------------|---|
| FULL | <p>Handling of Source Field Values:</p> <p>Without FULL, the following are removed from the source fields before the values are transferred:</p> <ul style="list-style-type: none"> ● leading zeros before the decimal point for fields of format N, P or I ● trailing zeros after the decimal point for fields of format N or P ● trailing blanks for fields of format A ● and leading binary zeros for fields of format B <p>For a numeric source field containing all zeros, one zero will be transferred. For example:</p> <pre>COMPRESS 'ABC ' 001 INTO #TARGET WITH DELIMITER '*' Content of #TARGET is: ABC*1</pre> <p>With FULL, the values of the source fields in their actual lengths will be transferred to the target field. In other words:</p> <ul style="list-style-type: none"> ● leading zeros before the decimal point for fields of format N, P or I ● trailing zeros after the decimal point for fields of format N or P ● and trailing blanks for fields of format A ● leading binary zeros for fields of format B <p>are displayed as entered. For example:</p> <pre>COMPRESS FULL 'ABC ' 001 INTO #TARGET WITH DELIMITER '*' Content of #TARGET is: ABC *001</pre> |
| <i>operand1</i> | <p>Source Fields:</p> <p>As <i>operand1</i>, you specify the fields whose contents are to be transferred.</p> <p>Note:</p> <p>If <i>operand1</i> is not of format A or B, its content is converted into alphanumeric representation before it is transferred. If necessary, the alphanumeric representation is truncated.</p> <p>If <i>operand1</i> is a time variable (format T), only the time component of the variable content is transferred, but not the date component.</p> |
| <i>operand2</i> | <p>Target Field:</p> <p>As <i>operand2</i>, you specify the field which is to receive the values of the source fields.</p> <p>If the target field is of format U (Unicode) and if a source field of format B is involved, the length of the sending binary field must be even.</p> |
| LEAVING SPACE | <p>Values in Target Field Separated by a Blank:</p> <p>If you use the COMPRESS statement without any further options, or if you specify LEAVING SPACE (which also applies by default), the values in the target field will be separated from one another by a blank.</p> |
| LEAVING NO SPACE | <p>Values in Target Field Not Separated:</p> <p>If you specify LEAVING NO SPACE, the values in the target field will not be separated from one another by a blank or any other character.</p> |

| Syntax Element | Description |
|--|---|
| <i>parameter</i> | <p>Print Mode/Date Format Parameters: As <i>parameter</i>, you can specify the session parameter PM or the session parameter DF:</p> |
| | <p>PM=I In order to support languages whose writing direction is from right to left, you can specify PM=I so as to transfer the value of <i>operand1</i> in inverse (right-to-left) direction to <i>operand2</i>. For example, as a result of the following statements, the content of #B would be ZYXABC:</p> <pre>MOVE 'XYZ' TO #A COMPRESS #A (PM=I) 'ABC' INTO #B LEAVING NO SPACE</pre> <p>Any trailing blanks in <i>operand1</i> will be removed (except if FULL is specified), then the value is reversed character by character and transferred to <i>operand2</i>.</p> |
| | <p>DF If <i>operand1</i> is a date variable, you can specify the session parameter DF as <i>parameter</i> for this variable.</p> |
| SUBSTRING (<i>operand1</i> , <i>operand3</i> , <i>operand4</i>) | <p>SUBSTRING Option: If <i>operand1</i> is of alphanumeric (A), Unicode (U) or binary (B) format, you can use the SUBSTRING option to transfer only a certain part of a source field. After the field name (<i>operand1</i>) you specify first the starting position (<i>operand3</i>) and then the length (<i>operand4</i>) of the field portion to be transferred.</p> |
| INTO SUBSTRING (<i>operand2</i> , <i>operand5</i> , <i>operand6</i>) | <p>INTO Clause: Also, you can use the SUBSTRING option in the INTO clause to transfer source values into a certain part of the target field.</p> <p>In both cases, the use of the SUBSTRING option in a COMPRESS statement corresponds to that in a MOVE statement. See the MOVE statement for details on the SUBSTRING option.</p> |
| WITH DELIMITERS | <p>Input Delimiter Character: If you wish the values in the target field to be separated from one another by a specific character, you use the DELIMITERS option.</p> <p>If you specify WITH DELIMITERS without <i>operand7</i>, the values will be separated by the input delimiter character as defined with the session parameter ID.</p> |
| WITH DELIMITERS <i>operand7</i> | <p>Specific Delimiter Character: If you specify WITH DELIMITERS <i>operand7</i>, the values will be separated by the character specified with <i>operand7</i>. <i>operand7</i> must be a single character. If <i>operand7</i> is a variable, it must be of format/length (A1) or (B1).</p> <p>If the target field is of format A or B, the format/length of the delimiter has to be (A1), (B1) or (U1).</p> <p>If the target field is of format U (Unicode), the format/length of the delimiter has to be (A1), (B2) or (U1).</p> |
| WITH ALL | <p>Handling of Delimiters: Without ALL, a delimiter is placed in the target field only between values actually transferred. For example:</p> <pre>COMPRESS 'A' ' ' 'C' ' ' INTO #TARGET WITH DELIMITERS '*' Content of #TARGET is: A*C</pre> <p>With ALL, a delimiter is also placed in the target field for each blank value that is not actually transferred. This means that the number of delimiters in the target field corresponds to the number of source fields minus 1. This may be useful, for example, if the content of the target field is to be separated again with a subsequent SEPARATE statement. For example:</p> <pre>COMPRESS 'A' ' ' 'C' ' ' INTO #TARGET WITH ALL DELIMITERS '*' Content of #TARGET is: A**C*</pre> |

Processing

A destination field of format B is handled like a destination field of format A.

The COMPRESS operation terminates when either all operands have been processed or the target field (*operand2*) is filled.

If the target field contains more positions than all operands combined, all remaining positions of *operand2* will be filled with blanks. If the target field is shorter, the value will be truncated.

If *operand2* is a dynamic variable, the COMPRESS operation terminates when all source operands have been processed. No truncation will be performed. The length of *operand2* after the COMPRESS operation will correspond to the combined length of the source operands. The current length of a dynamic variable can be ascertained by using the system variable *LENGTH.

Examples

This section covers the following topics:

- Example 1 - Compress
- Example 2 - Compress Leaving No Space
- Example 3 - Compress with Delimiter

Example 1 - Compress

```

** Example 'CMPEX1': COMPRESS
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 NAME
  2 FIRST-NAME
  2 MIDDLE-I
*
1 #COMPRESSED-NAME (A20)
END-DEFINE
*
LIMIT 4
READ EMPLOY-VIEW BY NAME
  COMPRESS FIRST-NAME MIDDLE-I NAME INTO #COMPRESSED-NAME
  DISPLAY NOTITLE
    FIRST-NAME MIDDLE-I NAME 5X #COMPRESSED-NAME
END-READ
*
END

```

Output of Program CMPEX1:

| FIRST-NAME | MIDDLE-I | NAME | #COMPRESSED-NAME |
|------------|----------|----------|-------------------|
| KEPA | | ABELLAN | KEPA ABELLAN |
| ROBERT | W | ACHIESON | ROBERT W ACHIESON |
| SIMONE | | ADAM | SIMONE ADAM |
| JEFF | H | ADKINSON | JEFF H ADKINSON |

Example 2 - Compress Leaving No Space

```

** Example 'CMPEX2': COMPRESS (with LEAVING NO SPACE)
*****
DEFINE DATA LOCAL
1 EMPL-VIEW VIEW OF EMPLOYEES
  2 NAME
  2 CURR-CODE (1)
  2 SALARY (1)
*
1 #CCSALARY (A20)
END-DEFINE
*
LIMIT 4
READ EMPL-VIEW BY NAME

  COMPRESS CURR-CODE (1) SALARY (1) INTO #CCSALARY
    LEAVING NO SPACE
  DISPLAY NOTITLE
    NAME CURR-CODE (1) SALARY (1) 5X #CCSALARY
END-READ
*
END

```

Output of Program CMPEX2:

| NAME | CURRENCY CODE | ANNUAL SALARY | #CCSALARY |
|----------|------------------|------------------|------------|
| ABELLAN | PTA | 1450000 | PTA1450000 |
| ACHIESON | UKL | 11300 | UKL11300 |
| ADAM | FRA | 159980 | FRA159980 |
| ADKINSON | USD | 34500 | USD34500 |

Example 3 - Compress with Delimiter

```

** Example 'CMPEX3': COMPRESS (with delimiter)
*****
DEFINE DATA LOCAL
1 EMPL-VIEW VIEW OF EMPLOYEES
  2 NAME
  2 CURR-CODE (1)
  2 SALARY (1)
*
1 #CCSALARY (A20)
END-DEFINE
*
LIMIT 4
READ EMPL-VIEW BY NAME

  COMPRESS CURR-CODE (1) SALARY (1) INTO #CCSALARY
    WITH DELIMITER '*'
  DISPLAY NOTITLE NAME CURR-CODE (1) SALARY (1) 5X #CCSALARY
END-READ
*
END

```

Output of Program CMPEX3:

| NAME | CURRENCY CODE | ANNUAL SALARY | #CCSALARY |
|----------|------------------|------------------|-------------|
| ABELLAN | PTA | 1450000 | PTA*1450000 |
| ACHIESON | UKL | 11300 | UKL*11300 |
| ADAM | FRA | 159980 | FRA*159980 |
| ADKINSON | USD | 34500 | USD*34500 |