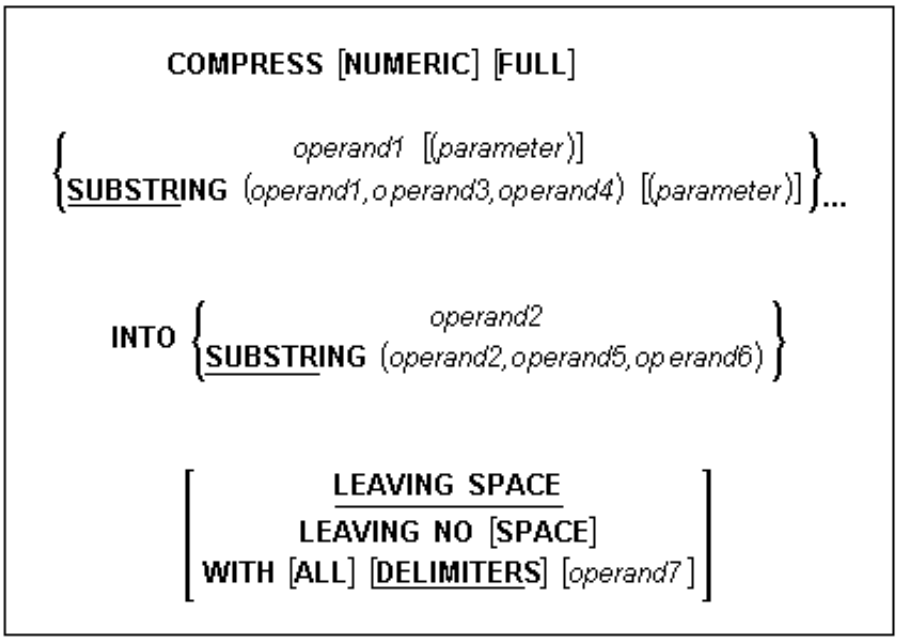
COMPRESS



Operand	Possible Structure					Possible Formats											Referencing Permitted	Dynamic Definition
Operand1	C	S	A	G	N	A	N	P	I	F	B	D	T			G	O	yes	no
Operand2		S				A												yes	yes
Operand3	C	S					N	P	I									yes	no
Operand4	C	S					N	P	I									yes	no
Operand5	C	S					N	P	I									yes	no
Operand6	C	S					N	P	I									yes	no
Operand7	C	S				A												yes	no

Related Statements: EXAMINE | SEPARATE

Function

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

Source Fields - operand1

As *operand1*, you specify the fields whose contents are to be transferred.

Note:
If operand1 is a time variable (format T), only the time component of the variable content is transferred, but not the date component.

Target Field - operand2

As *operand2*, you specify the field which is to receive the values of the source fields.

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.

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.

FULL

Without FULL, leading zeros (in numeric fields) and trailing blanks (in alphanumeric fields) are removed from the source fields before the values are transferred. For a numeric source field containing all zeros, one "0" will be transferred.

With FULL, the values of the source fields in their actual lengths - that is, including leading zeros and trailing blanks - will be transferred to the target field.

Examples:

```
1.    COMPRESS 'ABC '  001 INTO #TARGET WITH DELIMITER '*'
    Content of #TARGET is: ABC*1
2.    COMPRESS FULL 'ABC '  001 INTO #TARGET WITH DELIMITER '*'
    Content of #TARGET is: ABC *001
```

NUMERIC

This option determines how sign characters and decimal characters are to be handled:

- Without NUMERIC, decimal points and signs in numeric source values are suppressed before the values are transferred.
- With NUMERIC, decimal points and signs in numeric source values are also transferred to the target field.

Examples:

```
1.    COMPRESS -123  1.23 INTO #TARGET WITH DELIMITER '*'
    Content of #TARGET is: 123*123
2.    COMPRESS NUMERIC -123  1.23 INTO #TARGET WITH DELIMITER '*'
    Content of #TARGET is: -123*1.23
```

parameter

As parameter, you can specify the option "PM=I" or the session parameter DF:

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 *operand1* in inverse (right-to-left) direction to *operand2*.

For example, as a result of the following statements, the content of #B would be "ZYXABC":

```
MOVE 'XYZ' TO #A
COMPRESS #A (PM=I) 'ABC' INTO #B LEAVING NO SPACE
```

Any trailing blanks in *operand1* will be removed (except if FULL is specified), then the value is reversed character by character and transferred to *operand2*.

DF

If *operand1* is a date variable, you can specify the session parameter DF as *parameter* for this variable.

SUBSTRING

If *operand1* is of alphanumeric format, you can use the SUBSTRING option to transfer only a certain part of a source field.

Also, you can use the SUBSTRING option in the INTO clause to transfer source values into a certain part of the target field.

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.

WITH DELIMITER - operand7

If you wish the values in the target field to be separated from one another by a specific character, you use the DELIMITER option:

- If you specify WITH DELIMITER *operand7*, the values will be separated by the character specified with *operand7*. *Operand7* must be a single character. If *operand7* is a variable, it must be of format/length A1.
- If you specify WITH DELIMITERS without *operand7*, the values will be separated by the input delimiter character (as defined with the session parameter ID).

ALL

Without ALL, a delimiter is placed in the target field only between values actually transferred.

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.

Examples:

```
1.    COMPRESS 'A' ' ' 'C' ' ' INTO #TARGET WITH DELIMITER '*'
    Content of #TARGET is: A*C
```

```
2.    COMPRESS 'A' ' ' 'C' ' ' INTO #TARGET WITH ALL DELIMITERS '*'
    Content of #TARGET is: A**C*
```

Processing

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. For general information on DYNAMIC variables, see your Natural User's Guide.

Example 1


```
/* EXAMPLE 'CMPEX1S:' COMPRESS (STRUCTURED MODE)
/************************************************************
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
```

```
   FIRST-NAME      MIDDLE-I         NAME               #COMPRESSED-NAME
-------------------- -------- -------------------     --------------------

KEPA                          ABELLAN                 KEPA ABELLAN
ROBERT               W        ACHIESON                ROBERT W ACHIESON
SIMONE                        ADAM                    SIMONE ADAM
TIMMIE               D        ADKINSON                TIMMIE D ADKINSON
```

Equivalent reporting-mode example: See program CMPEX1R in library SYSEXRM.

Example 2

```
/* EXAMPLE 'CMPEX2': COMPRESS LEAVING NO SPACE
/************************************************************
LIMIT 4
READ EMPLOYEES BY NAME
 COMPRESS CURR-CODE (1) SALARY (1) INTO #CCSALARY (A20)
          LEAVING NO SPACE
 DISPLAY NOTITLE NAME CURR-CODE (1) SALARY (1) 5X #CCSALARY
/************************************************************
END
```

```
      NAME          CURRENCY   ANNUAL            #CCSALARY
                       CODE     SALARY
-------------------- -------- ----------     --------------------

ABELLAN               PTA         1450000     PTA1450000
ACHIESON              UKL           10500     UKL10500
ADAM                  FRA          159980     FRA159980
ADKINSON              USD           36000     USD36000
```

Example 3

```
/* EXAMPLE 'CMPEX3': COMPRESS WITH DELIMITER
/************************************************************
LIMIT 4
READ EMPLOYEES BY NAME
 COMPRESS CURR-CODE (1) SALARY (1) INTO #CCSALARY (A20)
          WITH DELIMITER '*'
 DISPLAY NOTITLE NAME CURR-CODE (1) SALARY (1) 5X #CCSALARY
/************************************************************
END
```

```
      NAME          CURRENCY    ANNUAL            #CCSALARY
                       CODE      SALARY
-------------------- -------- ----------     --------------------

ABELLAN                PTA         1450000     PTA*1450000
ACHIESON               UKL           10500     UKL*10500
ADAM                   FRA          159980     FRA*159980
ADKINSON               USD           36000     USD*36000
```

