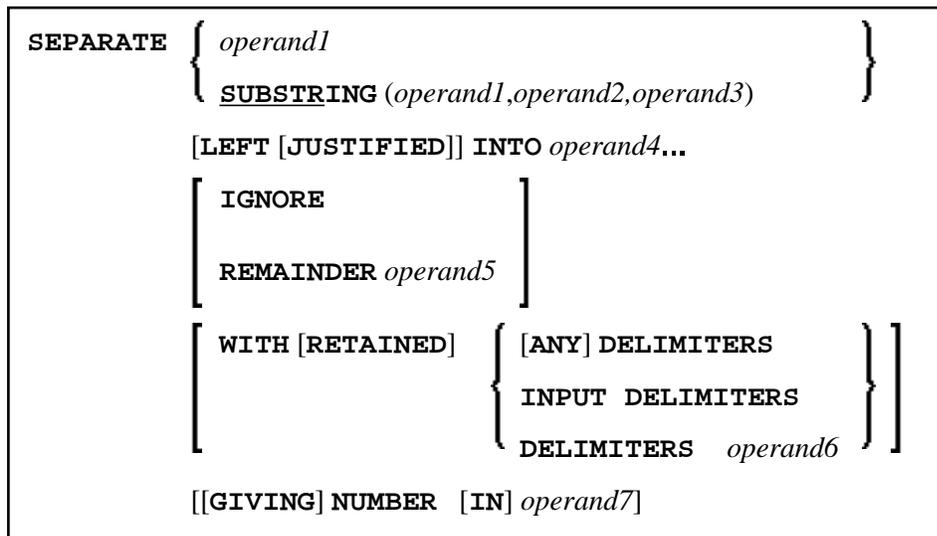


SEPARATE



Dieses Kapitel behandelt folgende Themen:

- Funktion
- Syntax-Beschreibung
- Beispiele

Eine Erläuterung der in dem Syntax-Diagramm verwendeten Symbole entnehmen Sie dem Abschnitt *Syntax-Symbole*.

Verwandte Statements: COMPRESS | COMPUTE | EXAMINE | MOVE | MOVE ALL | RESET

Gehört zur Funktionsgruppe: *Arithmetische Funktionen und Datenzuweisungen*

Funktion

Das Statement SEPARATE dient dazu, den Inhalt eines alphanumerischen oder binären Operanden auf zwei oder mehr alphanumerische oder binäre Operanden (oder auf mehrere Ausprägungen eines alphanumerischen oder binären Arrays) zu verteilen.

Syntax-Beschreibung

Operanden-Definitionstabelle:

Operand	Mögliche Struktur				Mögliche Formate								Referenzierung erlaubt	Dynam. Definition		
<i>operand1</i>	C	S			A	U					B				ja	nein
<i>operand2</i>	C	S					N	P	I	B*					ja	nein
<i>operand3</i>	C	S					N	P	I	B*					ja	nein
<i>operand4</i>		S	A	G	A	U				B					ja	ja
<i>operand5</i>		S			A	U				B					ja	ja
<i>operand6</i>	C	S			A	U				B					ja	nein
<i>operand7</i>		S					N	P	I						ja	ja

* Format B von *operand2* und *operand3* können nur mit einer Länge von kleiner gleich 4 benutzt werden.

Syntax-Element-Beschreibung:

<i>operand1</i>	<p>Ausgangsoperand:</p> <p><i>operand1</i> ist die alphanumerische/binäre Konstante oder Variable, deren Inhalt aufgeteilt werden soll.</p> <p>Nachgestellte Leerzeichen in <i>operand1</i> werden entfernt, bevor der Wert verarbeitet wird (auch wenn das Leerzeichen als Begrenzungszeichen verwendet wird; vgl. DELIMITERS-Option).</p>
SUBSTRING	<p>SUBSTRING-Option:</p> <p>Normalerweise wird der ganze Inhalt des Feldes aufgeteilt, und zwar vom Anfang des Feldes an.</p> <p>Die SUBSTRING-Option ermöglicht es Ihnen, nur einen bestimmten Teil des Feldes aufzuteilen. In der SUBSTRING-Klausel geben Sie nach dem Feldnamen (<i>operand1</i>) zunächst die erste Stelle (<i>operand2</i>) und dann die Länge (<i>operand3</i>) des Feldteils an, der aufgeteilt werden soll. Wenn z.B. ein Feld #A den Wert CONTRAPTION enthält, würde SUBSTRING(#A , 5 , 3) den Wert RAP enthalten.</p> <p>Normalerweise wird der ganze Inhalt des Feldes aufgeteilt, und zwar vom Anfang des Feldes an.</p> <p>Anmerkung: Wenn Sie <i>operand2</i> weglassen, wird ab Anfang des Feldes (Position 1) aufgeteilt. Wenn Sie <i>operand3</i> weglassen, wird ab der angegebenen Stelle (<i>operand2</i>) bis zum Ende des Feldes aufgeteilt.</p>

LEFT JUSTIFIED	<p>LEFT JUSTIFIED-Option:</p> <p>Diese Option bewirkt, dass den aufgeteilten Feldwertteilen vorangestellte Leerzeichen aus den Zieloperanden entfernt werden.</p>
<i>operand4</i>	<p>Zieloperand:</p> <p><i>operand4</i> enthält die Zieloperanden, die die Teile des Ausgangsoperanden aufnehmen sollen. Wird als Zieloperand ein Array verwendet, wird es Ausprägung für Ausprägung mit den übertragenen Feldwertteilen gefüllt.</p> <p>Die Anzahl der Zieloperanden entspricht der Anzahl der Begrenzungszeichen (einschließlich nachgestellter Begrenzungszeichen) in <i>operand1</i>, plus 1.</p> <p>Ist <i>operand4</i> eine dynamische Variable, kann deren Länge mit der SEPARATE-Operation geändert werden. Die aktuelle Länge einer dynamischen Variable kann mittels der Systemvariable *LENGTH ermittelt werden.</p> <p>Allgemeine Informationen zu dynamischen Variablen finden Sie im Abschnitt <i>Dynamische und große Variablen benutzen im Leitfaden zur Programmierung</i>.</p>
IGNORE / REMAINDER <i>operand5</i>	<p>IGNORE/REMAINDER-Option:</p> <p>Wenn Sie nicht genug Zieloperanden angeben, um alle Feldwertteile aufzunehmen, erhalten Sie eine entsprechende Fehlermeldung.</p> <p>Um dies zu vermeiden, haben Sie zwei Möglichkeiten:</p> <ul style="list-style-type: none"> ● Wenn Sie IGNORE angeben, ignoriert Natural es, falls nicht genügend Zieloperanden zur Aufnahme des Ausgangswertes vorhanden sind. ● Wenn Sie REMAINDER <i>operand5</i> angeben, wird der Teil des Ausgangswertes, für den keine Zieloperanden mehr zur Verfügung stehen, in <i>operand5</i> gestellt. Den Inhalt von <i>operand5</i> können Sie dann weiter verarbeiten, zum Beispiel in einem weiteren SEPARATE-Statement. <p>Siehe auch <i>Beispiel 3</i>.</p>
DELIMITERS	<p>DELIMITERS-Option:</p> <p>Siehe <i>Delimiters-Option</i> weiter unten.</p>

<p>WITH RETAINED DELIMITERS</p>	<p>WITH RETAINED DELIMITERS-Option:</p> <p>Normalerweise werden die Begrenzungszeichen selbst nicht mit in die Zieloperanden übertragen.</p> <p>Wenn Sie allerdings RETAINED angeben, wird jedes Begrenzungszeichen (d.h. entweder das mit dem Session-Parameter ID festgelegte Standard-Begrenzungszeichen und Leerzeichen oder die mit <i>operand6</i> angegebenen Zeichen) ebenfalls in einen Zieloperanden übertragen.</p> <p>Beispiel:</p> <p>Das folgende SEPARATE-Statement überträgt 150 nach #B, das Plus-Zeichen (+) nach #C und 30 nach #D:</p> <pre>... MOVE '150+30' TO #A SEPARATE #A INTO #B #C #D WITH RETAINED DELIMITER '+' ...</pre> <p>Siehe auch <i>Beispiel 3</i>.</p>
<p>GIVING NUMBER <i>operand7</i></p>	<p>GIVING NUMBER-Option</p> <p>Diese Option bewirkt, dass die Anzahl der Zieloperanden, die mit einem Wert gefüllt wurden (einschließlich der mit Leerzeichen gefüllten), in <i>operand7</i> ausgegeben wird. Die Anzahl, die Sie erhalten, errechnet sich aus der Anzahl der Delimiterzeichen plus 1.</p> <p>Wenn Sie die IGNORE-Option verwenden, enthält <i>operand7</i> maximal die Anzahl der Zieloperanden (<i>operand4</i>).</p> <p>Wenn Sie die REMAINDER-Option verwenden, enthält <i>operand7</i> maximal die Anzahl der Zieloperanden (<i>operand4</i>) plus <i>operand5</i>.</p>

DELIMITERS-Option:

<p> $\left[\text{WITH [RETAINED]} \left\{ \begin{array}{l} \text{[ANY] DELIMITERS} \\ \text{INPUT DELIMITERS} \\ \text{DELIMITERS } \textit{operand6} \end{array} \right\} \right]$ </p>
--

Begrenzungszeichen innerhalb von *operand1* bestimmen die Stellen, an denen der Wert geteilt werden soll.

- Falls Sie die DELIMITERS-Option nicht angeben (oder WITH ANY DELIMITER angeben), wird jedes Leerzeichen sowie jedes Zeichen, das weder ein Buchstabe noch eine Ziffer ist, als Begrenzungszeichen interpretiert.

- WITH INPUT DELIMITERS bedeutet, dass das mit dem Session-Parameter ID definierte Zeichen sowie das Leerzeichen als Begrenzungszeichen gelten.
- WITH DELIMITERS *operand6* bedeutet, dass jedes der angegebenen Zeichen (*operand6*) als Begrenzungszeichen interpretiert wird. Wenn *operand6* nachgestellte Leerzeichen enthält, werden diese ignoriert.

Beispiele

- Beispiel 1 — Verschiedene Beispiele für den Gebrauch des SEPARATE-Statements
- Beispiel 2 — SEPARATE-Statement bei einem Array
- Beispiel 3 — Gebrauch der Optionen REMAINDER/RETAINED

Beispiel 1 — Verschiedene Beispiele für den Gebrauch des SEPARATE-Statements

```

** Example 'SEPEX1': SEPARATE
*****
DEFINE DATA LOCAL
1 #TEXT1 (A6) INIT <'AAABBB'>
1 #TEXT2 (A7) INIT <'AAA BBB'>
1 #TEXT3 (A7) INIT <'AAA-BBB'>
1 #TEXT4 (A7) INIT <'A.B/C,D'>
1 #FIELD1A (A6)
1 #FIELD1B (A6)
1 #FIELD2A (A3)
1 #FIELD2B (A3)
1 #FIELD3A (A3)
1 #FIELD3B (A3)
1 #FIELD4A (A3)
1 #FIELD4B (A3)
1 #FIELD4C (A3)
1 #FIELD4D (A3)
1 #NBT (N1)
1 #DEL (A5)
END-DEFINE
*
WRITE NOTITLE 'EXAMPLE A (SOURCE HAS NO BLANKS)'
SEPARATE #TEXT1 INTO #FIELD1A #FIELD1B GIVING NUMBER #NBT
WRITE / '=' #TEXT1 5X '=' #FIELD1A 4X '=' #FIELD1B 4X '=' #NBT
*
WRITE NOTITLE /// 'EXAMPLE B (SOURCE HAS EMBEDDED BLANK)'
SEPARATE #TEXT2 INTO #FIELD2A #FIELD2B GIVING NUMBER #NBT
WRITE / '=' #TEXT2 4X '=' #FIELD2A 7X '=' #FIELD2B 7X '=' #NBT
*
WRITE NOTITLE /// 'EXAMPLE C (USING DELIMITER ''-'')'
SEPARATE #TEXT3 INTO #FIELD3A #FIELD3B WITH DELIMITER '-'
WRITE / '=' #TEXT3 4X '=' #FIELD3A 7X '=' #FIELD3B
*
MOVE ',/' TO #DEL
WRITE NOTITLE /// 'EXAMPLE D USING DELIMITER' '=' #DEL
*
SEPARATE #TEXT4 INTO #FIELD4A #FIELD4B
#FIELD4C #FIELD4D WITH DELIMITER #DEL

```

```

WRITE      /      '=' #TEXT4 4X '=' #FIELD4A 7X '=' #FIELD4B
           /              19X '=' #FIELD4C 7X '=' #FIELD4D
*
END

```

Ausgabe des Programms SEPEX1:

EXAMPLE A (SOURCE HAS NO BLANKS)

```
#TEXT1: AAABBB      #FIELD1A: AAABBB      #FIELD1B:           #NBT: 1
```

EXAMPLE B (SOURCE HAS EMBEDDED BLANK)

```
#TEXT2: AAA BBB      #FIELD2A: AAA          #FIELD2B: BBB          #NBT: 2
```

EXAMPLE C (USING DELIMITER '-')

```
#TEXT3: AAA-BBB      #FIELD3A: AAA          #FIELD3B: BBB
```

EXAMPLE D USING DELIMITER #DEL: ,/

```
#TEXT4: A.B/C,D      #FIELD4A: A.B          #FIELD4B: C
                        #FIELD4C: D          #FIELD4D:
```

Beispiel 2 — SEPARATE-Statement bei einem Array

```

** Example 'SEPEX2': SEPARATE (using array variable)
*****
DEFINE DATA LOCAL
1 #INPUT-LINE (A60) INIT <'VALUE1, VALUE2,VALUE3'>
1 #FIELD      (A20/1:5)
1 #NUMBER     (N2)
END-DEFINE
*
SEPARATE #INPUT-LINE LEFT JUSTIFIED INTO #FIELD (1:5)
          GIVING NUMBER IN #NUMBER
*
WRITE NOTITLE #INPUT-LINE //
              #FIELD (1) /
              #FIELD (2) /
              #FIELD (3) /
              #FIELD (4) /
              #FIELD (5) /
              #NUMBER
*
END

```

Ausgabe des Programms SEPEX2:

VALUE1, VALUE2,VALUE3

VALUE1
 VALUE2
 VALUE3

3

Beispiel 3 — Gebrauch der Optionen REMAINDER/RETAINED

```

** Example 'SEPEX3': SEPARATE (with REMAINDER, RETAIN option)
*****
DEFINE DATA LOCAL
1 #INPUT-LINE (A60) INIT <'VAL1, VAL2, VAL3,VAL4'>
1 #FIELD (A10/1:4)
1 #REM (A30)
END-DEFINE
*
WRITE TITLE LEFT 'INP:' #INPUT-LINE /
           '#FIELD (1)' 13T '#FIELD (2)' 25T '#FIELD (3)'
           37T '#FIELD (4)' 49T 'REMAINDER'
           / '-----' 13T '-----' 25T '-----'
           37T '-----' 49T '-----'
*
SEPARATE #INPUT-LINE INTO #FIELD (1:2)
           REMAINDER #REM WITH DELIMITERS ',,'
WRITE #FIELD(1) 13T #FIELD(2) 25T #FIELD(3) 37T #FIELD(4) 49T #REM
*
RESET #FIELD(*) #REM
SEPARATE #INPUT-LINE INTO #FIELD (1:2)
           IGNORE WITH DELIMITERS ',,'
WRITE #FIELD(1) 13T #FIELD(2) 25T #FIELD(3) 37T #FIELD(4) 49T #REM
*
RESET #FIELD(*) #REM
SEPARATE #INPUT-LINE INTO #FIELD (1:4) IGNORE
           WITH RETAINED DELIMITERS ',,'
WRITE #FIELD(1) 13T #FIELD(2) 25T #FIELD(3) 37T #FIELD(4) 49T #REM
*
RESET #FIELD(*) #REM
*
SEPARATE SUBSTRING(#INPUT-LINE,1,50) INTO #FIELD (1:4)
           IGNORE WITH DELIMITERS ',,'
WRITE #FIELD(1) 13T #FIELD(2) 25T #FIELD(3) 37T #FIELD(4) 49T #REM
*
END
    
```

Ausgabe des Programms SEPEX3:

```

INP: VAL1, VAL2, VAL3,VAL4
#FIELD (1) #FIELD (2) #FIELD (3) #FIELD (4) REMAINDER
-----
VAL1 VAL2 VAL3,VAL4
VAL1 VAL2
VAL1 , VAL2 ,
VAL1 VAL2 VAL3 VAL4
    
```