

REDEFINE

$\text{REDEFINE } \left\{ \begin{array}{l} \text{operand1} (\left\{ \begin{array}{l} nX \\ \text{operand2} \end{array} \right\} \dots) \\ \dots \end{array} \right\}$

Dieses Kapitel behandelt folgende Themen:

- Funktion
- Einschränkung
- Syntax-Beschreibung
- Beispiele

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

Funktion

Das Statement REDEFINE dient dazu, ein Feld zu redefinieren. Das Ergebnis der Neudefinition können eine oder mehrere Benutzervariablen sein.

Mit einem REDEFINE-Statement können Sie gleichzeitig mehrere Felder redefinieren.

Einschränkung

Dieses Statement gilt nur im Reporting Mode. Um ein Feld im Structured Mode zu redefinieren, verwenden Sie die REDEFINE-Klausel des DEFINE DATA-Statements.

Syntax-Beschreibung

Operanden-Definitionstabelle:

Operand	Mögliche Struktur	Mögliche Formate	Referenzierung erlaubt	Dynam. Definition
<i>operand1</i>	S A G	A U N P I F B D T L C	ja	nein
<i>operand2</i>	S A G	A N P I F B D T L C	ja	ja

Syntax-Element-Beschreibung:

REDEFINE <i>operand1</i> <i>operand2</i>	<p>Methode der Redefinition:</p> <p>Die Byte-Positionen von <i>operand1</i> werden unabhängig vom Format von links nach rechts redefiniert.</p> <p>Das Format von <i>operand2</i> muss nicht mit dem von <i>operand1</i> identisch sein. Die Byte-Positionen des neudefinierten Feldes müssen zu den im Feld enthaltenen Daten passen; wird beispielsweise ein alphanumerisches Feld als numerisch redefiniert, und enthält es entsprechend der Format-Spezifikation keine numerischen Daten, so kann die Verwendung des Feldes einen Programmabbruch zur Folge haben.</p> <p>Weitere Redefinition:</p> <p>Ein mit einem REDEFINE-Statement neudefiniertes Feld kann mit einem weiteren REDEFINE-Statement nochmals redefiniert werden.</p>
<i>nX</i>	<p>Füllbyte-Notation:</p> <p>Mit der Notation <i>nX</i> können Sie in der redefinierten Variable oder dem nachgestellten Feld <i>n</i> Füllbytes definieren. Nachgestellte Füllbytes müssen nicht unbedingt angegeben werden.</p>

Beispiele

- Beispiel 1 — REDEFINE-Statement
- Beispiel 2 — REDEFINE-Statement
- Beispiel 3 — REDEFINE-Statement
- Beispiel 4 — REDEFINE-Statement

Beispiel 1 — REDEFINE-Statement

Die Benutzervariable #A (Format/Länge A10) enthält den Wert 123ABCDEFGF.

```
REDEFINE #A (#A1(N3) #A2(A7))
```

#A1 erhält den Wert 123, #A2 den Wert ABCDEFG.

Beispiel 2 — REDEFINE-Statement

Die Benutzervariable #B (Format/Länge A10) enthält den hexadezimalen Wert 12345CC1C2C3C4C5C6C7.

```
REDEFINE #B (#B1(P4) #B2(A7))
```

#B1 erhält den hexadezimalen Wert 12345C, #B2 den hexadezimalen Wert C1C2C3C4C5C6C7.

```
REDEFINE #B (#BB1(B2)8X) or REDEFINE #B(#BB1(B2))
```

#BB1 erhält den hexadezimalen Wert 1234.

Der Wert in #BB1 ist "1234" (im Hexadecimalformat).

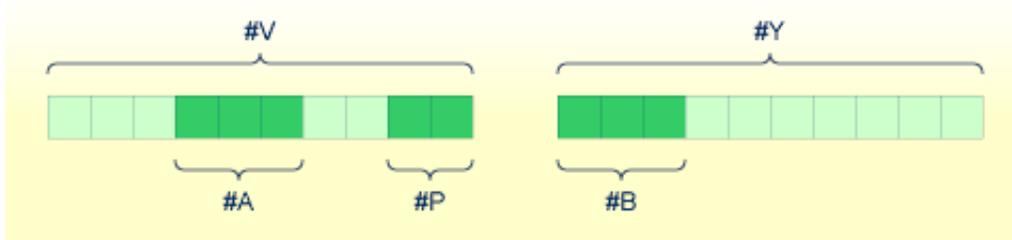
Anmerkung:

Beim Format P (gepackt numerisch) muss die Anzahl der benötigten Dezimalstellen angegeben werden. Die Anzahl der Bytes, die eine gepackte Zahl benötigt, lässt sich wie folgt berechnen:

Anzahl der Bytes = (Anzahl der Dezimalstellen + 1) / 2, auf ganze Bytes aufgerundet.

Beispiel 3 — REDEFINE-Statement

```
COMPUTE #V (N8.2) = #Y (N10) = ...
REDEFINE #V (3X #A(N3) 2X #P (N2)) #Y (#B(N3) 7X)
```



Beispiel 4 — REDEFINE-Statement

In diesem Beispiel wird die Systemvariable *DATN, die die Form YYYYMMDD hat, redefiniert und das Ergebnis in der Reihenfolge Tag/Monat/Jahr (DAY/MONTH/YEAR) in drei getrennte Felder geschrieben:

```
MOVE *DATN TO #DATINT (N8)
REDEFINE #DATINT (#YEAR (N4) #MONTH (N2) #DAY (N2))
DISPLAY NOTITLE #DATINT #DAY #MONTH #YEAR
END
```

Ausgabe:

```
#DATINT #DAY #MONTH #YEAR
-----
19950108 8 1 1995
```