REDEFINE

```
REDEFINE  { operand1 ( { nX        }   ) }
          {              { operand2 }... }...
```

This chapter covers the following topics:

- Function

- Restriction

- Syntax Description

- Examples

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

Function

The REDEFINE statement is used to redefine a field. The resulting definition may consist of one or more user-defined variables.

With one REDEFINE statement, several fields may be redefined.

Restriction

The REDEFINE statement is only valid in reporting mode. To redefine a field in structured mode, use the REDEFINE clause of the DEFINE DATA statement.

Syntax Description

Operand Definition Table:

Operand	Possible Structure			Possible Formats												Referencing Permitted	Dynamic Definition
operand1	S	A	G	A	U	N	P	I	F	B	D	T	L	C			yes	no
operand2	S	A	G	A		N	P	I	F	B	D	T	L	C			yes	yes

Syntax Element Description:

Syntax Element	Description
REDEFINE *operand1* *operand2*	**Method of Redefinition:** The byte positions of *operand1* are redefined from left to right regardless of format. The format of *operand2* may be different from the format of *operand1*. The bytes as specified in the REDEFINE statement must positionally match the data contained in the field being redefined. If an alphanumeric field is redefined as numeric and does not contain numeric data according to the format specification, an abnormal termination may result when it is used. **Further Redefinition:** Fields defined using a REDEFINE statement may be subsequently redefined with another REDEFINE statement.
*n*X	**Filler Notation:** The *n*X notation is used to denote filler bytes within the field/variable being redefined. Any trailing *n*X notation is optional.

Examples

- Example 1
- Example 2
- Example 3
- Example 4

Example 1

The user-defined variable #A (format/length A10) contains the value 123ABCDEFG.

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

The value in #A1 is 123. The value in #A2 is ABCDEFG.

Example 2

The user-defined variable #B (format/length A10) contains the value (shown in hexadecimal format) 12345CC1C2C3C4C5C6C7.

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

Example 3   **REDEFINE**

The value in `#B1` is `12345C` (in hexadecimal format).

The value in `#B2` is `C1C2C3C4C5C6C7` (in hexadecimal format).

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

The value in `#BB1` is `1234` (in hexadecimal format).

Note:
For packed data (Format P), the number of decimal positions required must be specified. The following formula can be used to determine the number of bytes that the packed number occupies:

```
Number of bytes = (number of decimal positions + 1) / 2,
```
rounded upwards to full bytes.

Example 3

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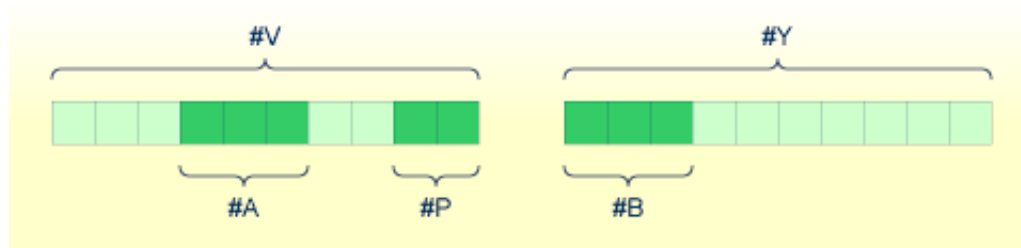

Example 4

This example redefines the value of the system variable `*DATN`, which is in the form `YYYYMMDD`, and displays the result as three separate fields in the order "day/month/year":

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

Output:

```
#DATINT  #DAY #MONTH #YEAR
  --------- ---- ------ -----

  19950108   8    1     1995
```