

# Definition von Array-Dimensionen

Die Definition von Array-Dimensionen ist im Statement `DEFINE DATA OBJECT` möglich und außerdem in der *variable-definition*-Option bei folgenden Statements: `DEFINE DATA LOCAL`, `DEFINE DATA INDEPENDENT`, `DEFINE DATA CONTEXT`, `DEFINE DATA OBJECT`.

Die *array-dimension-definition* hat die folgende Syntax:

```
{bound[:bound]},... 3
```

Dieses Kapitel behandelt folgende Themen:

- Funktion
- Syntax-Beschreibung

## Funktion

Mit der *array-dimension-definition* können Sie bei einer Array-Definition die Unter- und Obergrenze (*bound*) einer Dimension festlegen.

Sie können bis zu 3 Dimensionen für ein Array definieren.

Siehe auch *Arrays* im *Leitfaden zur Programmierung*.

## Syntax-Beschreibung

<b><i>bound</i></b>	<p>Als Grenze (<i>bound</i>) kann eines der folgenden Elemente verwendet werden:</p> <ul style="list-style-type: none"> <li>● eine numerische Ganzzahl-Konstante</li> <li>● eine vorher definierte Namens-Konstante</li> <li>● (bei Datenbank-Arrays) eine vorher definierte Benutzervariable</li> <li>● ein Stern (*); dieser definiert einen erweiterbaren <i>bound</i>, auch bekannt als X-Array.</li> </ul> <p>Wenn nur ein Bound angegeben ist, stellt der Wert die Obergrenze dar, und die Untergrenze wird als 1 angenommen.</p>
---------------------	---

### X-Arrays

Wenn mindestens eine Grenze (*bound*) in mindestens einer Dimension eines Arrays als erweiterbar angegeben wird, bezeichnet man dieses Array als X-Array (eXtensible Array). Nur eine Grenze (entweder oberer oder unterer) in einer Dimension kann erweiterbar sein, aber nicht beide. Mehrdimensionale Arrays können eine Mischung von konstanten und erweiterbaren Grenzen haben, z.B. `#a(1:100, 1:*)`.

Beispiel:

```
DEFINE DATA LOCAL
1 #ARRAY1(I4/1:10)
1 #ARRAY2(I4/10)
1 #X-ARRAY3(I4/1:*)
1 #X-ARRAY4(I4/*,1:5)
1 #X-ARRAY5(I4/*:10)
1 #X-ARRAY6(I4/1:10,100:* ,*:1000)
END-DEFINE
```

Die folgende Tabelle enthält eine Übersicht über die Grenzen der Arrays aus dem obigen Programm.

	Dimension 1		Dimension 2		Dimension 3	
	Untere Grenze	Obere Grenze	Untere Grenze	Obere Grenze	Untere Grenze	Obere Grenze
#ARRAY1	1	10	-	-	-	-
#ARRAY2	1	10	-	-	-	-
#X-ARRAY3	1	erweiterbar	-	-	-	-
#X-ARRAY4	1	erweiterbar	1	5	-	-
#X-ARRAY5	erweiterbar	10	-	-	-	-
#X-ARRAY6	1	10	100	erweiterbar	erweiterbar	1000

Beispiele für Array-Definitionen:

```
#ARRAY2(I4/10) /* a one-dimensional array with 10 occurrences (1:10)
#X-ARRAY4(I4/*,1:5) /* a two-dimensional array
#X-ARRAY6(I4/1:10,100:* ,*:1000) /* a three-dimensional array
```

### Variable Arrays in einer Parameter Data Area:

In einer Parameter Data Area können Sie ein Array mit einer variablen Anzahl von Ausprägungen angeben. Dies erfolgt mittels der Index-Notation 1:V.

Beispiel 1: #ARR01 (A5/1:V)

Beispiel 2: #ARR02 (I2/1:V,1:V)

Ein Parameter-Array, das eine variable Index-Notation 1:V enthält, kann nur redefiniert werden in der Länge

- seiner elementaren Feldlänge, wenn der Index 1:V ganz rechts steht, zum Beispiel:

```
#ARR(A6/1:V) kann bis zu einer Länge von 6 Bytes redefiniert werden;
#ARR(A6/1:2,1:V) kann bis zu einer Länge von 6 Bytes redefiniert werden;
#ARR(A6/1:2,1:3,1:V) kann bis zu einer Länge von 6 Bytes redefiniert werden.
```

- des Produkts der ganz rechts stehenden festen Ausprägungen und der elementaren Feldlänge, zum Beispiel:

#ARR(A6/1:V,1:2) kann bis zu einer Länge von  $2*6 = 12$  Bytes redefiniert werden;  
 #ARR(A6/1:V,1:3,1:2) kann bis zu einer Länge von  $3*2*6 = 36$  Bytes redefiniert werden;  
 #ARR(A6/1:2,1:V,1:3) kann bis zu einer Länge von  $3*6 = 18$  Bytes redefiniert werden.

Eine variable Index-Notation 1:V darf nicht in einem Redefinitionsblock verwendet werden.

Beispiel:

```
DEFINE DATA PARAMETER
  1 #ARR(A6/1:V)
  1 REDEFINE #ARR
    2 #R-ARR(A1/1:V) /* (1:V) is not allowed in a REDEFINE block
END-DEFINE
```

Da die Anzahl der Ausprägungen zur Kompilierungszeit nicht bekannt ist, darf es nicht mit der Index-Notation (\*) in den Statements INPUT, WRITE, READ WORK FILE und WRITE WORK FILE referenziert werden. Die Index-Notation (\*) kann entweder für alle Dimensionen oder für keine Dimension benutzt werden.

Gültige Beispiele:

```
#ARR01 (*)
#ARR02 (*,*)
#ARR01 (1)
#ARR02 (5,#FIELDX)
#ARR02 (1,1:3)
```

Ungültiges Beispiel:

```
#ARRAYY (1,*) /* not allowed
```

Um Laufzeitfehler zu vermeiden, sollte die maximale Anzahl der Ausprägungen eines solchen Arrays über einen anderen Parameter an das Subprogramm/die Subroutine übergeben werden. Als Alternative dazu können Sie die Systemvariable \*OCCURRENCE benutzen.

**Anmerkungen:**

1. Wenn eine einen Index 1:V enthaltende Parameter Data Area als eine (in einem DEFINE DATA LOCAL-Statement angegebene) Local Data Area benutzt wird, muss eine Variable mit Namen V als CONSTANT definiert worden sein.
2. In einem Dialog kann ein Index 1:V nicht in Zusammenhang mit BY VALUE benutzt werden.