X-Arrays

Wenn Sie ein normales Array definieren, müssen sie die Indexgrenzen und folglich die Anzahl der Ausprägungen für jede Dimension genau angeben. Zur Laufzeit existiert standardmäßig das vollständige Array-Feld; auf jede seiner Ausprägungen kann ohne zusätzliche Zuweisungsoperationen zugegriffen werden. Das Größen-Layout kann nicht mehr geändert werden; Sie können Feldausprägungen weder hinzufügen noch entfernen.

Zur Anwendungsentwicklungszeit kennen Sie wahrscheinlich nicht die genaue Anzahl der Ausprägungen eines Arrays. Sie möchten vielleicht aber die Größe eines Arrays zur Laufzeit ändern können.

Zu diesem Zweck können Sie ein sogenanntes X-Array (*eXtensible array* = erweiterbares Array) definieren. Die Größe eines X-Arrays kann zur Laufzeit geändert werden, wodurch Sie darin unterstützt werden, Ihren Hauptspeicher effizienter zu verwalten. Sie können z.B. eine große Anzahl von Array-Ausprägungen für einen kurzen Zeitraum benutzen und dann den Hauptspeicherplatz reduzieren, wenn die Anwendung den Array nicht mehr benutzt.

Dieses Kapitel behandelt folgende Themen:

- Definition
- Speicherverwaltung von X-Arrays
- Speicherverwaltung von X-Gruppen-Arrays
- X-Array referenzieren
- Parameter-Übertragung mit X-Arrays
- Parameter-Übertragung mit X-Group-Arrays
- X-Array mit dynamischen Variablen
- Unter- und Obergrenze eines Arrays

Definition

Ein X-Array ist ein Array, dessen Anzahl an Ausprägungen zur Kompilierungszeit nicht bekannt ist. Ein X-Array kann nur in einem `DEFINE DATA`-Statement definiert werden, wenn Sie einen Stern (*) für mindestens eine Grenze von wenigstens einer Dimension des Arrays angeben. Der Stern (*) in der Grenzen-Definition verweist darauf, dass die entsprechende Grenze verlängerbar ist. Nur eine Grenze — entweder die obere oder die untere — kann verlängert werden, aber nicht beide.

Ein X-Array kann immer dann definiert werden, wenn ein (fester) Array definiert werden kann, d.h. auf jeder Ebene oder sogar als indizierte Gruppe. Es kann nicht verwendet werden, um auf MU-PE-Felder zuzugreifen. Ein mehrdimensionales Array kann eine Mischung aus konstanten und verlängerbaren Grenzen haben.

Beispiel:

```
DEFINE DATA LOCAL
1 #X-ARR1 (A5/1:*)          /* lower bound is fixed at 1, upper bound is variable
1 #X-ARR2 (A5/*)            /* shortcut for (A5/1:*)
1 #X-ARR3 (A5/*:100)        /* lower bound is variable, upper bound is fixed at 100
1 #X-ARR4 (A5/1:10,1:*)     /* 1st dimension has a fixed index range with (1:10)
END-DEFINE                  /* 2nd dimension has fixed lower bound 1 and variable upper bound
```

Speicherverwaltung von X-Arrays

Die Ausprägungen eines X-Arrays müssen ausdrücklick zugewiesen werden, bevor auf sie zugegriffen werden kann. Sie können für die Statements EXPAND, RESIZE und REDUCE die Anzahl der Ausprägungen für jede Dimension ändern.

Die Anzahl der Dimension des X-Arrays (1, 2 oder 3 Dimensionen) kann aber nicht geändert werden.

Beispiel:

```
DEFINE DATA LOCAL
1 #X-ARR(I4/10:*)
END-DEFINE
EXPAND ARRAY #X-ARR TO (10:10000)
/* #X-ARR(10) to #X-ARR(10000) are accessible
WRITE *LBOUND(#X-ARR)                  /* is 10
   *UBOUND(#X-ARR)                     /* is 10000
   *OCCURRENCE(#X-ARR)                 /* is 9991
#X-ARR(*) := 4711                      /* same as #X-ARR(10:10000) := 4711
/* resize array from current lower bound=10 to upper bound =1000
RESIZE ARRAY #X-ARR TO (*:1000)
/* #X-ARR(10) to #X-ARR(1000) are accessible
/* #X-ARR(1001) to #X-ARR(10000) are released
WRITE *LBOUND(#X-ARR)                  /* is 10
   *UBOUND(#X-ARR)                     /* is 1000
   *OCCURRENCE(#X-ARR)                 /* is 991
/* release all occurrences
REDUCE ARRAY #X-ARR TO  0
WRITE *OCCURRENCE(#X-ARR)              /* is 0
```

Speicherverwaltung von X-Gruppen-Arrays

Wenn Sie Ausprägungen der X-Gruppen-Arrays erhöhen oder reduzieren möchten, müssen Sie zwischen unabhängigen und abhängigen Dimensionen unterscheiden.

Eine Dimension, die direkt (nicht weitergegeben) für einen X-Gruppen-Array angegeben wird, ist *unabhängig*.

Eine Dimension, die nicht *direkt* für einen X-Gruppen-Array angegeben, sondern weitergegeben wird, ist *abhängig*.

Nur die unabhängigen Dimensionen können in den Statements EXPAND, RESIZE und REDUCE geändert werden. Die Dimensionen müssen unter Verwendung des entsprechenden Namens des X-Gruppen-Arrays, zu dem diese Dimension als unabhängige Dimension gehört, geändert werden.

Beispiel - Unabhängige/abhängige Dimensionen:

```
DEFINE  DATA LOCAL
1 #X-GROUP-ARR1(1:*)                 /* (1:*)
  2 #X-ARR1    (I4)                  /* (1:*)
  2 #X-ARR2    (I4/2:*)              /* (1:*,2:*)
  2 #X-GROUP-ARR2                    /* (1:*)
    3 #X-ARR3  (I4)                  /* (1:*)
    3 #X-ARR4  (I4/3:*)              /* (1:*,3:*)
    3 #X-ARR5  (I4/4:*, 5:*)         /* (1:*,4:*,5:*)
END-DEFINE
```

Die folgende Tabelle zeigt, ob die Dimensionen in dem obengezeigten Programm unabhängig oder abhängig sind.

Name	Abhängige Dimension	Unabhängige Dimension
#X-GROUP-ARR1		(1:*)
#X-ARR1	(1:*)	
#X-ARR2	(1:*)	(2:*)
#X-GROUP-ARR2	(1:*)	
#X-ARR3	(1:*)	
#X-ARR4	(1:*)	(3:*)
#X-ARR5	(1:*)	(4:*,5:*)

Als Index-Notation für die abhängige Dimension ist entweder ein einzelner Stern (*), ein mit einem Stern definierter Bereich oder die Definition der Ober- und Untergrenze zulässig. Diese Angabe soll darauf verweisen, dass die Grenzen der abhängigen Dimension so bleiben müssen wie sie sind und nicht geändert werden können.

Die Ausprägungen der abhängigen Dimensionen können nur durch Manipulation der entsprechenden Array-Gruppen geändert werden.

```
EXPAND ARRAY #X-GROUP-ARR1 TO (1:11)       /* #X-ARR1(1:11) are allocated
                                           /* #X-ARR3(1:11) are allocated
EXPAND ARRAY #X-ARR2 TO (*:*, 2:12)        /* #X-ARR2(1:11, 2:12) are allocated
EXPAND ARRAY #X-ARR2 TO (1:*, 2:12)        /* same as before
EXPAND ARRAY #X-ARR2 TO (*  , 2:12)        /* same as before
EXPAND ARRAY #X-ARR4 TO (*:*, 3:13)        /* #X-ARR4(1:11, 3:13) are allocated
EXPAND ARRAY #X-ARR5 TO (*:*, 4:14, 5:15)  /* #X-ARR5(1:11, 4:14, 5:15) are allocated
```

Die EXPAND-Statements können in beliebiger Reihenfolge kodiert werden.

Die folgende Verwendung des EXPAND-Statements ist nicht zulässig, da die Arrays nur abhängige Dimensionen haben.

```
EXPAND ARRAY #X-ARR1 TO ...
EXPAND ARRAY #X-GROUP-ARR2 TO ...
EXPAND ARRAY #X-ARR3 TO ...
```

X-Array referenzieren

Die Ausprägungen eines X-Arrays müssen über ein EXPAND- oder RESIZE-Statement zugewiesen werden, bevor sie aufgerufen werden können.

In der Regel gilt: der Versuch, eine nicht existierende X-Array-Ausprägung zu adressieren führt zu einem Laufzeitfehler. Bei einigen Statements verursacht der Zugriff auf ein nicht verwirklichtes X-Array-Feld jedoch keinen Fehler, wenn alle Ausprägungen eines X-Arrays mit der kompletten Bereichsnotation referenziert werden, zum Beispiel #X-ARR(*). Dies gilt für

- Parameter, die in einem CALL-Statement benutzt werden,
- Parameter, die bei CALLNAT, PERFORM, SEND EVENT oder OPEN DIALOG benutzt werden, wenn sie als OPTIONAL definiert sind,
- Source-Felder, die in einem COMPRESS-Statement benutzt werden,
- Ausgabefelder, die mit einem PRINT-Statement angegeben werden,
- Felder, die in einem RESET-Statement referenziert werden.

Wenn individuelle Ausprägungen eines nicht verwirklichten X-Arrays in einem dieser Statements referenziert werden, wird eine entsprechende Fehlermeldung ausgegeben.

Beispiel:

```
DEFINE DATA LOCAL
1 #X-ARR (A10/1:*) /* X-array only defined, but not allocated
END-DEFINE
RESET #X-ARR(*)    /* no error, because complete field referenced with (*)
RESET #X-ARR(1:3)  /* runtime error, because individual occurrences (1:3) are referenced
END
```

Die Stern-Notation (*) in einer Array-Referenz steht für den kompletten Bereich einer Dimension. Wenn das Array ein X-Array ist, dann steht der Stern für den Indexbereich der aktuell zugewiesenen Unter- und Obergrenze, die mit *LBOUND und *UBOUND festgelegt wird.

Parameter-Übertragung mit X-Arrays

Als Parameter benutzte X-Arrays werden im Hinblick auf die Überprüfung folgender Elemente wie Konstanten-Arrays behandelt:

- Format
- Länge
- Dimension

  oder

- Anzahl der Ausprägungen

Außerdem können X-Array-Parameter auch die Anzahl der Ausprägungen ändern, wenn Sie die Statements `RESIZE`, `REDUCE` oder `EXPAND` benutzen. Die RESIZE-Funktion eines X-Array-Parameters ist von drei Faktoren abhängig:

- der Art der benutzten Parameter-Übertragung, d.h. By Reference oder By Value
- der Definition des Caller oder des X-Array-Parameters
- dem Typ des weitergegebenen X-Array-Bereichs (kompletter Bereich oder Unterbereich)

Die folgenden Tabellen zeigen, wann ein `EXPAND`, `RESIZE` oder `REDUCE` eines X-Array-Parameters zulässig ist.

Beispiel mit CALL By Value

CALLER	**PARAMETER**		
	Statisch	Variable (1:V)	`X-Array`
Statisch	Nein	Nein	Ja
X-Array, Unterbereich, z.B. `CALLNAT...#XA(1:5)`	Nein	Nein	Ja
X-Array, kompletter Bereich, z.B. `CALLNAT...#XA(*)`	Nein	Nein	Ja

CALL By Reference/CALL By Value Result

CALLER	**PARAMETER**			

	Statisch	Variable (1:V)	X-Array mit einer festen Untergrenze, kompletter Bereich, z.B.
			
			`DEFINE DATA PARAMETER`
			`1 #PX (A10/1:*)`
				X-Array mit einer festen Obergrenze, kompletter Bereich, z.B.
				
				`DEFINE DATA PARAMETER`
				`1 #PX (A10/*:1)`

Statisch	Nein	Nein	Nein	Nein
X-Array, Unterbereich, z.B. `CALLNAT...#XA(1:5)`	Nein	Nein	Nein	Nein

X-Array mit einer festen Untergrenze, kompletter Bereich, z.B.
 	
`DEFINE DATA LOCAL`
`1 #XA(A10/1:*)`
`...`
`CALLNAT...#XA(*)`
	Nein	Nein	Ja	Nein

X-Array mit einer festen Obergrenze, kompletter Bereich, z.B.
 	
`DEFINE DATA LOCAL`
`1 #XA(A10/*:1)`
`...`
`CALLNAT...#XA(*)`
	Nein	Nein	Nein	Ja

Parameter-Übertragung mit X-Group-Arrays

Die Deklaration eines X-Group-Arrays impliziert, dass jedes Element der Gruppe dieselben Werte für die Ober- und Untergrenze hat. Aus diesem Grund kann die Anzahl der Ausprägungen von abhängigen Felddimensionen eines X-Group-Arrays nur geändert werden, wenn der Gruppenname des X-Group-Arrays mit dem Statement `RESIZE`, `REDUCE` und `EXPAND` angegeben wird (siehe *Speicherverwaltung von X-Gruppen-Arrays* oben).

Bestandteile von X-Group-Arrays können als Parameter an X-Group-Arrays übertragen werden, die in einer Parameter Data Area definiert sind. Die Gruppenstrukturen des Aufrufers und des Aufgerufenen müssen nicht unbedingt identisch sein. Ein `RESIZE`, `REDUCE` und `EXPAND`, das vom Aufgerufenen gemacht wird, ist nur möglich solange das X-Group-Array des Aufrufers gleich bleibt.

Beispiel - Elemente eines X-Group Array als Parameter übertragen:

Programm:

```
DEFINE DATA LOCAL
1 #X-GROUP-ARR1(1:*)                    /* (1:*)
   2 #X-ARR1 (I4)                       /* (1:*)
   2 #X-ARR2 (I4)                       /* (1:*)
1 #X-GROUP-ARR2(1:*)                    /* (1:*)
   2 #X-ARR3 (I4)                       /* (1:*)
   2 #X-ARR4 (I4)                       /* (1:*)
END-DEFINE
...
CALLNAT ... #X-ARR1(*) #X-ARR4(*)
...
END
```

Subprogramm:

```
DEFINE DATA PARAMETER
1 #X-GROUP-ARR(1:*)                     /* (1:*)
   2 #X-PAR1 (I4)                       /* (1:*)
   2 #X-PAR2 (I4)                       /* (1:*)
END-DEFINE
...
RESIZE ARRAY #X-GROUP-ARR to (1:5)
...
END
```

Das RESIZE-Statement ist im Subprogramm nicht möglich. Das Ergebnis wäre eine uneinheitliche Anzahl von Ausprägungen der Felder, die in den X-Group-Arrays des Programms definiert sind.

X-Array mit dynamischen Variablen

Ein X-Array mit dynamischen Variablen kann zugewiesen werden, indem man zuerst die Anzahl der Ausprägungen mit Hilfe des EXPAND-Statements angibt und dann den zuvor zugewiesenen Array-Ausprägungen einen Wert zuweist.

Beispiel:

```
DEFINE DATA LOCAL
 1 #X-ARRAY(A/1:*)  DYNAMIC
END-DEFINE
EXPAND  ARRAY   #X-ARRAY TO (1:10)
  /* allocate #X-ARRAY(1) to #X-ARRAY(10) with zero length.
  /* *LENGTH(#X-ARRAY(1:10)) is zero
#X-ARRAY(*) := 'abc'
  /* #X-ARRAY(1:10) contains 'abc',
  /* *LENGTH(#X-ARRAY(1:10)) is 3
EXPAND  ARRAY   #X-ARRAY TO (1:20)
  /* allocate #X-ARRAY(11) to #X-ARRAY(20) with zero length
  /* *LENGTH(#X-ARRAY(11:20)) is zero
#X-ARRAY(11:20) := 'def'
  /* #X-ARRAY(11:20) contains 'def'
  /* *LENGTH(#X-ARRAY(11:20)) is 3
```

Unter- und Obergrenze eines Arrays

Die Systemvariablen *LBOUND und *UBOUND enthalten die aktuelle Unter- und Obergrenze eines Arrays für die angegebene(n) Dimension(en) (1, 2 oder 3).

Wenn keine Ausprägungen eines X-Arrays zugewiesen worden sind, ist der Aufruf von *LBOUND oder *UBOUND für die variablen Indexgrenzen nicht definiert, d.h. für die durch einen Stern (*) in der Indexdefinition dargestellten Grenzen, und führt zu einem Laufzeitfehler. Um einen Laufzeitfehler zu vermeiden, kann *OCCURRENCE benutzt werden, um auf Null-Ausprägungen zu überprüfen, bevor *LBOUND oder *UBOUND ausgewertet wird:

Beispiel:

```
IF *OCCURRENCE (#A) NE 0  AND  *UBOUND(#A) < 100 THEN ...
```