

FOR

FOR	<i>operand1</i>	[{	[:]=	}]	<i>operand2</i>
			{	EQ	}		
			{	FROM	}		
			[{	TO	}	<i>operand3</i> [[STEP] <i>operand4</i>]
			{	THRU	}		
							<i>statement ...</i>
END-FOR							<i>(structured mode only)</i>
[LOOP]							<i>(reporting mode only)</i>

Dieses Kapitel behandelt folgende Themen:

- Funktion
- Syntax-Beschreibung
- Beispiel — FOR-Statement

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

Verwandte Statements: REPEAT | ESCAPE

Gehört zur Funktionsgruppe: *Schleifenverarbeitung*

Funktion

Mit dem Statement `FOR` wird eine Verarbeitungsschleife ausgelöst und gleichzeitig die Anzahl der Schleifendurchläufe gesteuert.

Konsistenzprüfung

Bevor die `FOR`-Schleife zum erstenmal durchlaufen wird, wird geprüft, ob die Werte der Operanden konsistent sind (d.h. ob es möglich ist, dass durch wiederholtes Addieren von *operand4* zu *operand2* der Wert von *operand3* erreicht werden kann); ist dies nicht der Fall, wird die `FOR`-Schleife nicht durchlaufen (ohne dass eine Fehlermeldung ausgegeben wird; Ausnahme: wenn der `STEP`-Wert Null ist).

Syntax-Beschreibung

Operanden-Definitionstabelle:

Operand	Mögliche Struktur			Mögliche Formate										Referenzierung erlaubt	Dynam. Definition		
<i>operand1</i>		S			N	P	I	F								ja	ja
<i>operand2</i>	C	S		N	N	P	I	F								ja	nein
<i>operand3</i>	C	S		N	N	P	I	F								ja	nein
<i>operand4</i>	C	S		N	N	P	I	F								ja	nein

Syntax-Element-Beschreibung:

<i>operand1</i>	Schleifenkontrollvariable (<i>operand1</i>) und Ausgangswert (<i>operand2</i>): <i>operand1</i> kann ein Datenbankfeld oder eine Benutzervariable sein und steuert die Anzahl der Schleifendurchläufe. Der nach dem Schlüsselwort FROM angegebene Wert (<i>operand2</i>) wird der Kontrollvariablen als Ausgangswert zugeordnet, bevor die Verarbeitungsschleife das erste Mal durchlaufen wird. Dieser Ausgangswert erhöht <i>operand2</i> wird. sich mit jedem Schleifendurchlauf um den Wert des mit STEP angegebenen <i>operand4</i> (bzw. vermindert sich, wenn <i>operand4</i> einen negativen Wert hat).
<i>operand2</i>	Die Kontrollvariable kann während der Ausführung der Verarbeitungsschleife referenziert werden, um den aktuellen Wert der Kontrollvariablen zu erhalten.
<i>operand3</i>	TO-Wert: Die Verarbeitungsschleife wird beendet, sobald <i>operand1</i> einen Wert enthält, der größer ist als der Wert von <i>operand3</i> (oder kleiner, falls der STEP-Wert negativ ist).
<i>operand4</i>	STEP-Wert: Der Wert von <i>operand4</i> kann positiv oder negativ sein. Ist kein Wert angegeben, wird ein Wert von +1 angenommen. Je nach dem Vorzeichen des STEP-Wertes wird die Vergleichsoperation für <i>operand3</i> auf <i>größer als</i> oder <i>kleiner als</i> gesetzt, wenn die Verarbeitungsschleife zum erstenmal durchlaufen wird. <i>operand4</i> darf nicht Null (0) sein.
END-FOR	Das für Natural reservierte Wort END-FOR muss zum Beenden des FOR-Statements benutzt werden.

Beispiel — FOR-Statement

```

** Example 'FOREX1S': FOR (structured mode)
*****
DEFINE DATA LOCAL
1 #INDEX (I1)
1 #ROOT (N2.7)
END-DEFINE
*
FOR #INDEX 1 TO 5
  COMPUTE #ROOT = SQRT (#INDEX)
  WRITE NOTITLE '=' #INDEX 3X '=' #ROOT

```

```
END-FOR
*
SKIP 1
FOR #INDEX 1 TO 5 STEP 2
  COMPUTE #ROOT = SQRT (#INDEX)
  WRITE '=' #INDEX 3X '=' #ROOT
END-FOR
*
END
```

Ausgabe des Programms FOREX1S:

```
#INDEX: 1 #ROOT: 1.0000000
#INDEX: 2 #ROOT: 1.4142135
#INDEX: 3 #ROOT: 1.7320508
#INDEX: 4 #ROOT: 2.0000000
#INDEX: 5 #ROOT: 2.2360679

#INDEX: 1 #ROOT: 1.0000000
#INDEX: 3 #ROOT: 1.7320508
#INDEX: 5 #ROOT: 2.2360679
```

Äquivalentes Reporting-Mode-Beispiel: FOREX1R.