

# AT BREAK

## Structured Mode-Syntax

```
[AT] BREAK [(r)] [OF] operand1 [/n/]  
    statement ...  
END-BREAK
```

## Reporting Mode-Syntax

```
[AT] BREAK [(r)] [OF] operand1 [/n/]  
{  
    statement  
    DO statement... DOEND  
}
```

Dieses Kapitel behandelt folgende Themen:

- Funktion
- Syntax-Beschreibung
- Gruppenwechsel auf mehreren Ebenen
- Beispiele

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

Verwandte Statements: ACCEPT/REJECT | AT START OF DATA | AT END OF DATA | BACKOUT TRANSACTION | BEFORE BREAK PROCESSING | DELETE | END TRANSACTION | FIND | GET | GET SAME | GET TRANSACTION DATA | HISTOGRAM | LIMIT | PASSW | PERFORM BREAK PROCESSING | READ | RETRY | STORE | UPDATE

Gehört zur Funktionsgruppe: *Datenbankzugriffe und Datenbankänderungen*

---

## Funktion

Das Statement AT BREAK dient dazu, in einer mit FIND, READ, HISTOGRAM, SORT oder READ WORK FILE initiierten Verarbeitungsschleife eine an einen automatischen Gruppenwechsel geknüpfte Verarbeitung anzugeben. Mit dem AT BREAK-Statement können Sie ein oder mehrere andere Statements angeben, die jedesmal ausgeführt werden sollen, wenn der Wert eines bestimmten Feldes (Kontrollfeld) sich ändert.

Die automatische Gruppenwechsel-Verarbeitung funktioniert folgendermaßen: Unmittelbar nachdem ein Datensatz in der Verarbeitungsschleife gelesen worden ist, wird das Kontrollfeld geprüft. Wenn im Vergleich zum vorangegangenen Datensatz eine Wertänderung festgestellt wird, dann werden die im AT BREAK-Statement-Block enthaltenen Statements ausgeführt. Dies gilt nicht für den ersten Datensatz in der

Verarbeitungsschleife. Zusätzlich wird am Ende der Verarbeitungsschleife (weil alle Datensätze gelesen sind oder wegen eines `ESCAPE BOTTOM`-Statements) eine letzte Ausführung der `AT BREAK`-Statement-Block enthaltenen Statements veranlasst.

Weitere Informationen siehe *Automatische Gruppenwechsel-Verarbeitung im Leitfaden zur Programmierung*.

Ein `AT BREAK`-Statement-Block wird nur ausgeführt, wenn das Objekt, das den Statement-Block enthält, zu dem Zeitpunkt, zu dem die Gruppenwechsel-Bedingung auftritt, aktiv ist.

Es ist auch möglich, mit einer `AT BREAK`-Verarbeitung eine weitere Verarbeitungsschleife zu initiieren. Die Schleife muss allerdings innerhalb der `AT BREAK`-Verarbeitung wieder beendet werden.

Dieses Statement ist nicht prozedural (das heißt, seine Ausführung hängt von einem Ereignis ab, nicht davon, wo im Programm es steht).

Natural-Systemfunktionen können in Verbindung mit einem `AT BREAK`-Statement benutzt werden, siehe *Natural-Systemfunktionen für Verarbeitungsschleifen* in der *Systemfunktionen-Dokumentation* und *Beispiel für Systemfunktionen mit AT BREAK-Statement* im *Leitfaden zur Programmierung*.

Weitere Informationen siehe auch den Abschnitt *AT BREAK-Statement* im *Leitfaden zur Programmierung*. Darin werden Themen behandelt wie zum Beispiel:

- *Gruppenwechsel basierend auf einem Datenbankfeld*
- *Gruppenwechsel basierend auf einer Benutzervariablen*

## Syntax-Beschreibung

Operanden-Definitionstabelle:

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

Syntax-Element-Beschreibung:

<i>(r)</i>	<p><b>Referenzierungsnotation:</b> Ein AT BREAK-Statement wird zum letztenmal ausgeführt, wenn eine mit FIND, READ, READ WORK FILE, HISTOGRAM oder SORT initiierte Verarbeitungsschleife beendet wird. Normalerweise bezieht sich das AT BREAK-Statement hierbei auf die äußerste aktive Schleife.</p> <p>Wollen Sie, dass sich die abschließende AT BREAK-Verarbeitung auf eine andere offene Schleife bezieht (die Schleife, in der das AT BREAK-Statement steht, oder eine äußere Schleife), so verwenden Sie hierzu die Notation (<i>r</i>), wobei <i>r</i> das Statement-Label bzw. die Sourcecode-Zeilenummer des betreffenden schleifeninitiiierenden Statements ist.</p> <p><b>Beispiel:</b></p> <pre>0110 ... 0120 READ ... 0130   FIND ... 0140     FIND ... 0150       AT BREAK ... 0160         FIND ... 0170           END-FIND 0180             END-BREAK 0190               END-FIND 0200                 END-FIND 0210                   END-READ 0220 ...</pre> <p>In diesem Beispiel bezieht sich die abschließende AT BREAK-Bedingung auf die in Zeile 0120 initiierte READ-Schleife. Es wäre auch möglich, sie an eine der in Zeile 0130 bzw. 0140 initiierten FIND-Schleifen zu knüpfen, nicht jedoch an die in Zeile 0160 initiierte.</p> <p>Soll eine ganze Hierarchie von AT BREAK-Statements sich auf eine andere als die aktive Schleife beziehen, so müssen Sie die Notation (<i>r</i>) bei dem ersten AT BREAK-Statement angeben; sie bezieht sich dann auch auf alle innerhalb der Hierarchie folgenden AT BREAK-Statements.</p>
<i>operand1</i>	<p><b>Kontrollfeld:</b> In der Regel wird als Kontrollfeld ein Datenbankfeld verwendet. Sie können aber auch eine Benutzervariable nehmen, müssen diese allerdings vor der Gruppenwechsel-Verarbeitung definiert haben (siehe BEFORE BREAK PROCESSING-Statement). Sie können auch eine bestimmte Ausprägung eines Arrays als Kontrollfeld verwenden.</p>
<i>/n/</i>	<p>Sie haben auch die Möglichkeit, einen Teil eines Feldes zum Kontrollfeld zu machen:</p> <p>Mit der Notation <i>/n/</i> geben Sie an, dass nur die ersten <i>n</i> Stellen (von links nach rechts) des Feldes als Kontrollfeld dienen sollen, d.h. das AT BREAK-Statement wird nur ausgeführt, wenn der Wert der ersten <i>n</i> Stellen sich ändert. Diese Möglichkeit besteht allerdings nur bei Feldern, die das Format A, B, N oder P haben.</p> <p>Ein AT BREAK-Statement wird immer dann ausgeführt, wenn ein Gruppenwechsel stattfindet, das heißt, wenn der Wert des Kontrollfeldes sich ändert. Es wird außerdem ausgeführt, nachdem alle Datensätze in der Schleife, auf die sich das AT BREAK-Statement bezieht, verarbeitet worden sind.</p>

<b>END-BREAK</b>	Das reservierte Natural-Wort END-BREAK muss zum Beenden des AT BREAK-Statements benutzt werden.
------------------	---

## Gruppenwechsel auf mehreren Ebenen

Innerhalb einer Verarbeitungsschleife in demselben Programm-Modul können Sie mehrere AT BREAK-Statements verwenden. Damit schaffen Sie eine Hierarchie von AT BREAK-Statements, und zwar unabhängig davon, ob die AT BREAK-Statements unmittelbar aufeinander folgen oder zwischen ihnen noch andere Statements stehen. Das erste AT BREAK-Statement befindet sich auf der untersten Ebene der Hierarchie, jedes weitere auf einer nächsthöheren.

Für jede Schleife können Sie in einer Schleife eine eigene AT BREAK-Hierarchie aufbauen.

### Beispiel:

Structured Mode:	Reporting Mode:
<pre> FIND ...   AT BREAK   ...   END-BREAK   AT BREAK   ...   END-BREAK   AT BREAK   ...   END-BREAK   AT BREAK   ...   END-BREAK END-FIND ... </pre>	<pre> FIND ...   AT BREAK   DO   ...   DOEND   AT BREAK   DO   ...   DOEND ... </pre>

Bei einem Gruppenwechsel auf einer bestimmten Ebene werden auch alle AT BREAK-Statements auf jeweils untergeordneten Ebenen der Hierarchie ausgeführt, unabhängig davon, ob im Kontrollfeld einer unteren Ebene ebenfalls ein Gruppenwechsel stattgefunden hat.

Der Übersichtlichkeit halber empfiehlt es sich, die einzelnen AT BREAK-Statements einer Hierarchie unmittelbar aufeinanderfolgen zu lassen.

Siehe auch *Beispiel 3* und den Abschnitt *Gruppenwechsel auf mehreren Ebenen* im *Leitfaden zur Programmierung*.

## Beispiele

- Beispiel 1 — AT BREAK
- Beispiel 2 — AT BREAK mit der Notation /n/
- Beispiel 3 — AT BREAK mit Gruppenwechseln auf mehreren Ebenen

Weitere Beispiele für AT BREAK siehe *Systemfunktionen in Verarbeitungsschleifen*, Beispiele ATBEX3 und ATBEX4.

## Beispiel 1 — AT BREAK

```

** Example 'ATBEX1S': AT BREAK (structured mode)
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 CITY
  2 COUNTRY
  2 NAME
END-DEFINE
*
LIMIT 10
READ EMPLOY-VIEW BY CITY
AT BREAK OF CITY
  SKIP 1
  END-BREAK
  DISPLAY NOTITLE CITY (IS=ON) COUNTRY (IS=ON) NAME
END-READ
*
END

```

Ausgabe des Programms ATBEX1S:

CITY	COUNTRY	NAME
AIKEN	USA	SENKO
AIX EN OTHE	F	GODEFROY
AJACCIO		CANALE
ALBERTSLUND	DK	PLOUG
ALBUQUERQUE	USA	HAMMOND ROLLING FREEMAN LINCOLN
ALFRETON	UK	GOLDBERG
ALICANTE	E	GOMEZ

Äquivalentes Reporting-Mode-Beispiel: ATBEX1R.

## Beispiel 2 — AT BREAK mit der Notation /n/

```

** Example 'ATBEX2': AT BREAK (with /n/ notation)
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 DEPT
  2 NAME
END-DEFINE
*
LIMIT 10
READ EMPLOY-VIEW BY DEPT STARTING FROM 'A'
AT BREAK OF DEPT /4/
  SKIP 1
  END-BREAK

```

```

    DISPLAY NOTITLE DEPT NAME
END-READ
*
END

```

Ausgabe des Programms ATBEX2:

```

DEPARTMENT          NAME
  CODE
-----
ADMA01      JENSEN
ADMA01      PETERSEN
ADMA01      MORTENSEN
ADMA01      MADSEN
ADMA01      BUHL
ADMA02      HERMANSEN
ADMA02      PLOUG
ADMA02      HANSEN

COMP01      HEURTEBISE
COMP01      TANCHOU

```

### Beispiel 3 — AT BREAK mit Gruppenwechseln auf mehreren Ebenen

```

** Example 'ATBEX5S': AT BREAK (multiple break levels) (structured mode)
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 CITY
  2 DEPT
  2 NAME
  2 LEAVE-DUE
1 #LEAVE-DUE-L (N4)
END-DEFINE
*
LIMIT 5
FIND EMPLOY-VIEW WITH CITY = 'PHILADELPHIA' OR = 'PITTSBURGH'
      SORTED BY CITY DEPT
      MOVE LEAVE-DUE TO #LEAVE-DUE-L
      DISPLAY CITY (IS=ON) DEPT (IS=ON) NAME #LEAVE-DUE-L
/*
AT BREAK OF DEPT
  WRITE NOTITLE /
    T*DEPT OLD(DEPT) T*#LEAVE-DUE-L SUM(#LEAVE-DUE-L) /
END-BREAK
AT BREAK OF CITY
  WRITE NOTITLE
    T*CITY OLD(CITY) T*#LEAVE-DUE-L SUM(#LEAVE-DUE-L) //
END-BREAK
END-FIND
*
END

```

Ausgabe des Programms ATBEX5:

```

          CITY          DEPARTMENT          NAME          #LEAVE-DUE-L
          CODE
-----
PHILADELPHIA          MGMT30          WOLF-TERROINE          11

```

		MACKARNESS	27
	MGMT30		38
	TECH10	BUSH	39
		NETTLEFOLDS	24
	TECH10		63
PHILADELPHIA			101
PITTSBURGH	MGMT10	FLETCHER	34
	MGMT10		34
PITTSBURGH			34

Äquivalentes Reporting-Mode-Beispiel: ATBEX5R.