

# Gruppenwechsel

Dieses Kapitel beschreibt, wie die Ausführung eines Statements von einem Gruppenwechsel abhängig gemacht werden kann, und wie Gruppenwechsel für die Auswertung von Natural-Systemfunktionen benutzt werden können.

Folgende Themen werden behandelt:

- Verwendung von Gruppenwechseln
  - AT BREAK-Statement
  - Automatische Gruppenwechsel-Verarbeitung
  - Beispiel für Systemfunktionen in einem AT BREAK-Statement
  - Weiteres Beispiel für AT BREAK-Statement
  - BEFORE BREAK PROCESSING-Statement
  - Beispiel für BEFORE BREAK PROCESSING-Statement
  - Programmabhängige Gruppenwechsel-Verarbeitung — das PERFORM BREAK PROCESSING-Statement
  - Beispiel für PERFORM BREAK PROCESSING-Statement
- 

## Verwendung von Gruppenwechseln

Ein Gruppenwechsel (Break) findet statt, wenn der Wert eines Kontrollfeldes sich ändert.

Die Ausführung von Statements kann von einem solchen Gruppenwechsel abhängig gemacht werden.

Ein Gruppenwechsel kann auch zur Auswertung von Natural-Systemfunktionen verwendet werden.

Systemfunktionen werden im Abschnitt *Systemvariablen und Systemfunktionen* behandelt. Genauere Beschreibungen der verfügbaren Systemfunktionen System finden Sie in der *Systemfunktionen*-Dokumentation.

## AT BREAK-Statement

Mit dem Statement AT BREAK können Sie eine Verarbeitung angeben, die immer dann ausgeführt werden soll, wenn ein Gruppenwechsel erfolgt, d.h. wenn der Wert eines Kontrollfeldes, das Sie im AT BREAK-Statement angeben, sich ändert. Als Kontrollfeld können Sie ein Datenbankfeld oder eine Benutzervariable verwenden.

In diesem Abschnitt werden folgende Themen behandelt:



```

                COUNT(SALARY(1)) 'RECORDS FOUND' /
END-BREAK
/*
AT END OF DATA
    WRITE 'TOTAL (ALL RECORDS):' T*SALARY(1) TOTAL(SALARY(1))
END-ENDDATA
END-READ
END

```

Im obigen Programm wird das erste WRITE-Statement ausgeführt, wenn der Wert des Feldes CITY sich ändert.

Im AT BREAK-Statement werden die Systemfunktionen OLD, AVER und COUNT ausgewertet (und in dem WRITE-Statement ausgegeben).

In dem AT END OF DATA-Statement wird die Systemfunktion TOTAL ausgewertet.

Das Programm ATBREX01 erzeugt folgende Ausgabe:

```

Page          1                                04-12-14  14:07:26

  CITY                NAME                POSITION                SALARY
-----
AIKEN          SENKO                PROGRAMMER                31500
A I K E N                AVERAGE:                31500
      1 RECORDS FOUND

ALBUQUERQ     HAMMOND                SECRETARY                22000
ALBUQUERQ     ROLLING                MANAGER                34000
ALBUQUERQ     FREEMAN                MANAGER                34000
ALBUQUERQ     LINCOLN                ANALYST                41000
A L B U Q U E R Q U E                AVERAGE:                32750
      4 RECORDS FOUND

TOTAL (ALL RECORDS):                162500

```

## Gruppenwechsel basierend auf einer Benutzervariablen

Auch eine Benutzervariable kann als Kontrollfeld in einem AT BREAK-Statement verwendet werden.

Im folgenden Programm wird die Benutzervariable #LOCATION als Kontrollfeld verwendet.

```

** Example 'ATBREX02': AT BREAK OF (with user-defined variable and
**                               in conjunction with BEFORE BREAK PROCESSING)
*****
DEFINE DATA LOCAL
1 MYVIEW VIEW OF EMPLOYEES
  2 CITY
  2 COUNTRY
  2 JOB-TITLE
  2 SALARY (1:1)
*
1 #LOCATION (A20)
END-DEFINE
*

```

```

READ (5) MYVIEW BY CITY WHERE COUNTRY = 'USA'
  BEFORE BREAK PROCESSING
    COMPRESS CITY 'USA' INTO #LOCATION
  END-BEFORE
  DISPLAY #LOCATION 'POSITION' JOB-TITLE 'SALARY' SALARY (1)
  /*
  AT BREAK OF #LOCATION
    SKIP 1
  END-BREAK
END-READ
END

```

Ausgabe des Programms ATBREX02:

Page 1

04-12-14 14:08:36

| #LOCATION       | POSITION   | SALARY |
|-----------------|------------|--------|
| AIKEN USA       | PROGRAMMER | 31500  |
| ALBUQUERQUE USA | SECRETARY  | 22000  |
| ALBUQUERQUE USA | MANAGER    | 34000  |
| ALBUQUERQUE USA | MANAGER    | 34000  |
| ALBUQUERQUE USA | ANALYST    | 41000  |

## Gruppenwechsel auf mehreren Ebenen

Mit der Notation */n/* können Sie, wie oben erläutert, den Teil eines Feldes zum Kontrollfeld eines Gruppenwechsels machen. Sie können auch mehrere AT BREAK-Statements miteinander kombinieren, wobei bei einem Gruppenwechsel ein ganzes Feld und bei einem anderen ein Teil dieses Feldes Kontrollfeld ist.

In diesem Fall muss der übergeordnete Gruppenwechsel (ganzes Feld) vor dem untergeordneten (Teil des Feldes) angegeben werden, d.h. im ersten AT BREAK-Statement muss das ganze Feld, im zweiten das Teilfeld als Kontrollfeld angegeben werden.

Das folgende Beispielprogramm zeigt dies anhand des Feldes DEPT und den ersten 4 Stellen dieses Feldes (DEPT /4/).

```

** Example 'ATBREX03': AT BREAK OF (two statements in combination)
*****
DEFINE DATA LOCAL
1 MYVIEW VIEW OF EMPLOYEES
  2 NAME
  2 JOB-TITLE
  2 DEPT
  2 SALARY (1:1)
  2 CURR-CODE (1:1)
END-DEFINE
*
READ MYVIEW BY DEPT STARTING FROM 'SALE40' ENDING AT 'TECH10'
  WHERE SALARY(1) GT 47000 AND CURR-CODE(1) = 'USD'
/*
  AT BREAK OF DEPT
    WRITE '*** LOWEST BREAK LEVEL ***' /
  END-BREAK
  AT BREAK OF DEPT /4/
    WRITE '*** HIGHEST BREAK LEVEL ***'

```

```

END-BREAK
/*
  DISPLAY DEPT NAME 'POSITION' JOB-TITLE
END-READ
END

```

Ausgabe des Programms ATBEX03:

Page 1 04-12-14 14:09:20

| DEPARTMENT<br>CODE          | NAME   | POSITION |
|-----------------------------|--------|----------|
| TECH05                      | HERZOG | MANAGER  |
| TECH05                      | LAWLER | MANAGER  |
| TECH05                      | MEYER  | MANAGER  |
| *** LOWEST BREAK LEVEL ***  |        |          |
| TECH10                      | DEKKER | DBA      |
| *** LOWEST BREAK LEVEL ***  |        |          |
| *** HIGHEST BREAK LEVEL *** |        |          |

Im folgenden Programm wird jedesmal, wenn sich der Wert des Feldes DEPT ändert, eine Leerzeile ausgegeben; und jedesmal, wenn sich der Wert in den ersten 4 Stellen von DEPT ändert, wird über die Systemfunktion COUNT die Anzahl der verarbeiteten Datensätze ermittelt.

```

** Example 'ATBEX04': AT BREAK OF (two statements in combination)
*****
DEFINE DATA LOCAL
1 MYVIEW VIEW OF EMPLOYEES
  2 DEPT
  2 REDEFINE DEPT
    3 #GENDEP (A4)
  2 NAME
  2 SALARY (1)
END-DEFINE
*
WRITE TITLE '** PERSONS WITH SALARY > 30000, SORTED BY DEPARTMENT **' /
LIMIT 9
READ MYVIEW BY DEPT FROM 'A' WHERE SALARY(1) > 30000
  DISPLAY 'DEPT' DEPT NAME 'SALARY' SALARY(1)
/*
  AT BREAK OF DEPT
    SKIP 1
  END-BREAK
  AT BREAK OF DEPT /4/
    WRITE COUNT(SALARY(1)) 'RECORDS FOUND IN:' OLD(#GENDEP) /
  END-BREAK
END-READ
END

```

Ausgabe des Programms ATBEX04:

```

** PERSONS WITH SALARY > 30000, SORTED BY DEPARTMENT **

```

| DEPT   | NAME   | SALARY |
|--------|--------|--------|
| ADMA01 | JENSEN | 180000 |

|        |           |        |
|--------|-----------|--------|
| ADMA01 | PETERSEN  | 105000 |
| ADMA01 | MORTENSEN | 320000 |
| ADMA01 | MADSEN    | 149000 |
| ADMA01 | BUHL      | 642000 |
| ADMA02 | HERMANSEN | 391500 |
| ADMA02 | PLOUG     | 162900 |
| ADMA02 | HANSEN    | 234000 |

8 RECORDS FOUND IN: ADMA

|        |            |        |
|--------|------------|--------|
| COMP01 | HEURTEBISE | 168800 |
|--------|------------|--------|

1 RECORDS FOUND IN: COMP

## Automatische Gruppenwechsel-Verarbeitung

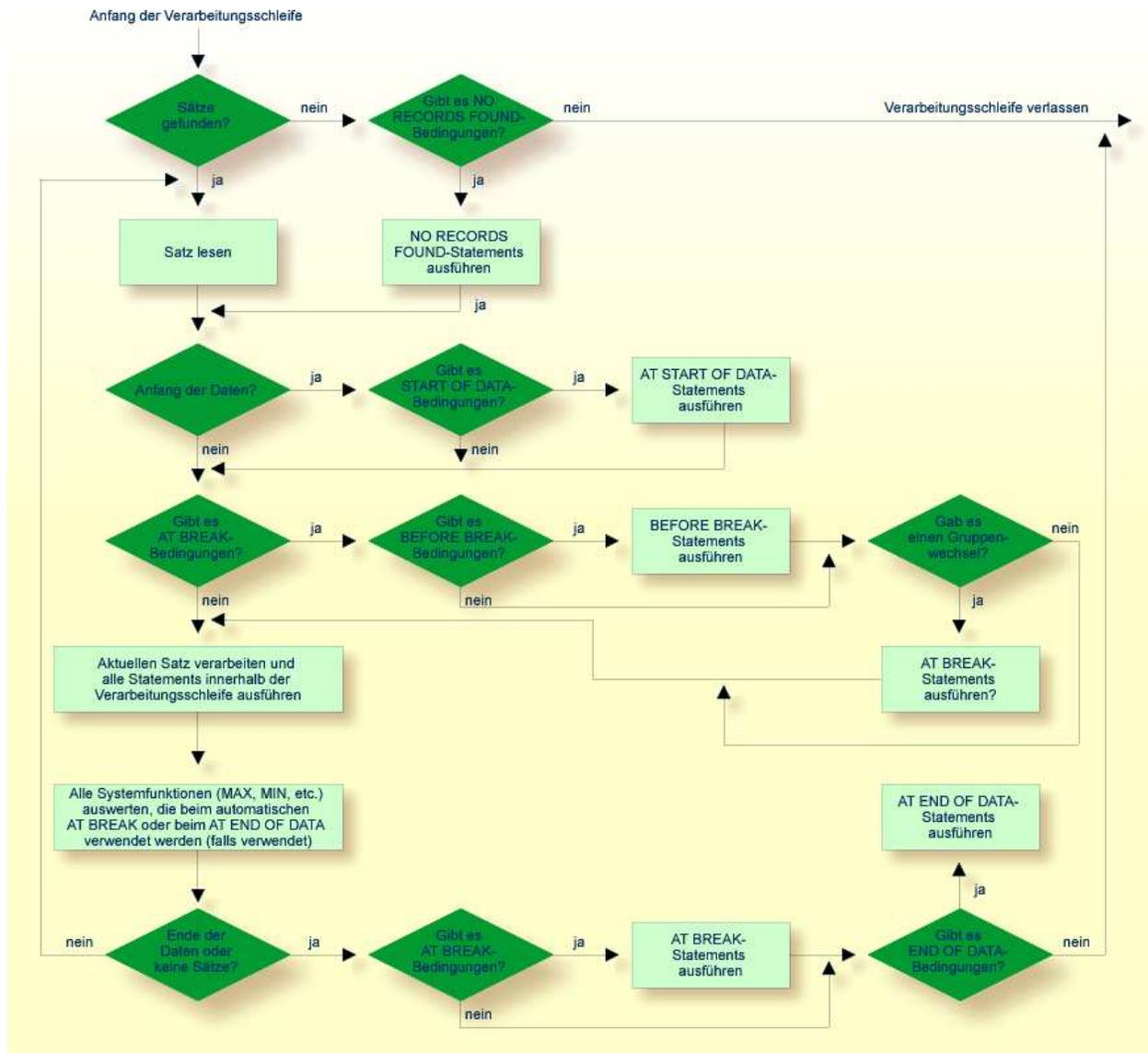
Automatische Gruppenwechsel-Verarbeitung ist für eine Verarbeitungsschleife aktiv, die ein AT BREAK-Statement enthält. Dies gilt für die folgenden Statements:

- FIND
- READ
- HISTOGRAM
- SORT
- READ WORK FILE

Hierbei wird der Wert des im AT BREAK-Statement angegebenen Kontrollfeldes nur bei den Datensätzen überprüft, die die WITH- und WHERE-Auswahlkriterien der Verarbeitungsschleife erfüllen.

Natural-Systemfunktionen (AVER, MAX, MIN usw.) werden für jeden Datensatz ausgewertet, nachdem alle in der Verarbeitungsschleife enthaltenen Statements ausgeführt worden sind. Datensätze, die aufgrund des WHERE-Kriteriums nicht verarbeitet werden, werden bei der Auswertung der Systemfunktionen nicht berücksichtigt.

Die Abbildung auf der folgenden Seite veranschaulicht den Verarbeitungsablauf eines automatischen Gruppenwechsels.



## Beispiel für Systemfunktionen in einem AT BREAK-Statement

Das folgende Beispiel zeigt die Verwendung der Natural-Systemfunktionen OLD, MIN, AVER, MAX, SUM und COUNT in einem AT BREAK-Statement (und der Systemfunktion TOTAL in einem AT END OF DATA-Statement).

```

** Example 'ATBREX05': AT BREAK OF (with system functions)
*****
DEFINE DATA LOCAL
1 MYVIEW VIEW OF EMPLOYEES
  2 NAME
  2 CITY
  2 SALARY (1:1)
  2 CURR-CODE (1:1)
END-DEFINE
*
LIMIT 3

```



## Beispiel für BEFORE BREAK PROCESSING-Statement

```

** Example 'BEFORX01': BEFORE BREAK PROCESSING
*****
DEFINE DATA LOCAL
1 MYVIEW VIEW OF EMPLOYEES
  2 NAME
  2 FIRST-NAME
  2 SALARY (1:1)
  2 BONUS (1:1,1:1)
*
1 #INCOME (P11)
END-DEFINE
*
LIMIT 5
READ MYVIEW BY NAME FROM 'B'
  BEFORE BREAK PROCESSING
    COMPUTE #INCOME = SALARY(1) + BONUS(1,1)
  END-BEFORE
/*
  DISPLAY NOTITLE NAME FIRST-NAME (AL=10)
    'ANNUAL/INCOME' #INCOME 'SALARY' SALARY(1) (LC==) /
    '+ BONUS' BONUS(1,1) (IC=+)
  AT BREAK OF #INCOME
    WRITE T*#INCOME '-'(24)
  END-BREAK
END-READ
END

```

Ausgabe des Programms BEFORX01:

| NAME     | FIRST-NAME | ANNUAL<br>INCOME | SALARY<br>+ BONUS |
|----------|------------|------------------|-------------------|
| BACHMANN | HANS       | 56800 =          | 52800<br>+4000    |
| BAECKER  | JOHANNES   | 81000 =          | 74400<br>+6600    |
| BAECKER  | KARL       | 52650 =          | 48600<br>+4050    |
| BAGAZJA  | MARJAN     | 152700 =         | 129700<br>+23000  |
| BAILLET  | PATRICK    | 198500 =         | 188000<br>+10500  |

## Programmabhängige Gruppenwechsel-Verarbeitung — das PERFORM BREAK PROCESSING-Statement

Bei automatischer Gruppenwechsel-Verarbeitung werden die im AT BREAK-Block angegebenen Statements jedesmal ausgeführt, wenn sich der Wert des angegebenen Kontrollfeldes ändert, und zwar unabhängig von der Position des AT BREAK-Statements in der Verarbeitungsschleife.

Mit einem PERFORM BREAK PROCESSING-Statement können Sie selbst festlegen, wo in einer Verarbeitungsschleife eine Gruppenwechsel-Verarbeitung ausgeführt werden soll. Das PERFORM BREAK PROCESSING-Statement wird dann ausgeführt, wenn es im Verarbeitungsablauf des Programms angetroffen wird.

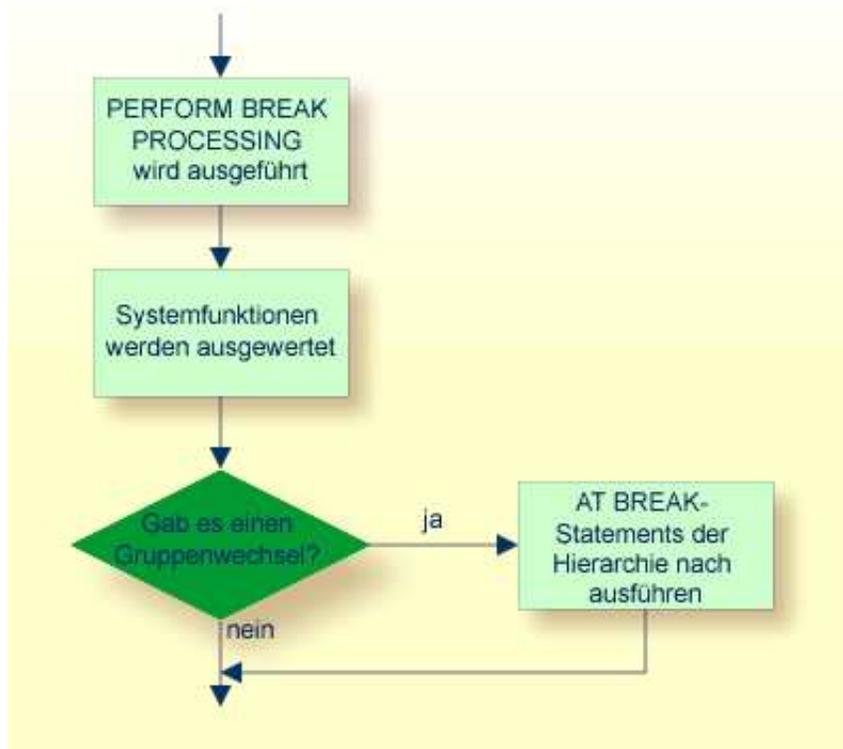
Unmittelbar nach dem PERFORM BREAK PROCESSING-Statement geben Sie einen oder mehrere AT BREAK-Statement-Blöcke an:

```
...  
PERFORM BREAK PROCESSING  
  AT BREAK OF field1  
    statements  
  END-BREAK  
  AT BREAK OF field2  
    statements  
  END-BREAK  
...
```

Wenn ein PERFORM BREAK PROCESSING-Statement ausgeführt wird, prüft Natural, ob ein Gruppenwechsel stattgefunden hat, d.h. ob der Wert des angegebenen Kontrollfeldes sich geändert hat; ist dies der Fall, dann werden die angegebenen Statements ausgeführt.

Bei PERFORM BREAK PROCESSING werden Systemfunktionen ausgewertet, *bevor* Natural prüft, ob ein Gruppenwechsel stattgefunden hat.

Die folgende Abbildung zeigt den logischen Ablauf einer programmabhängigen Gruppenwechsel-Verarbeitung:



# Beispiel für PERFORM BREAK PROCESSING-Statement

```

** Example 'PERFBX01': PERFORM BREAK PROCESSING (with BREAK option
**                               in IF statement)
*****
DEFINE DATA LOCAL
1 MYVIEW VIEW OF EMPLOYEES
  2 NAME
  2 DEPT
  2 SALARY (1:1)
*
1 #CNTL      (N2)
END-DEFINE
*
LIMIT 7
READ MYVIEW BY DEPT
  AT BREAK OF DEPT          /* <- automatic break processing
  SKIP 1
  WRITE 'SUMMARY FOR ALL SALARIES      '
    'SUM:'   SUM(SALARY(1))
    'TOTAL:' TOTAL(SALARY(1))
  ADD 1 TO #CNTL
END-BREAK
/*
IF SALARY (1) GREATER THAN 100000 OR BREAK #CNTL
  PERFORM BREAK PROCESSING /* <- user-initiated break processing
  AT BREAK OF #CNTL
    WRITE 'SUMMARY FOR SALARY GREATER 100000'
      'SUM:'   SUM(SALARY(1))
      'TOTAL:' TOTAL(SALARY(1))
  END-BREAK
END-IF
/*
IF SALARY (1) GREATER THAN 150000 OR BREAK #CNTL
  PERFORM BREAK PROCESSING /* <- user-initiated break processing
  AT BREAK OF #CNTL
    WRITE 'SUMMARY FOR SALARY GREATER 150000'
      'SUM:'   SUM(SALARY(1))
      'TOTAL:' TOTAL(SALARY(1))
  END-BREAK
END-IF
DISPLAY NAME DEPT SALARY(1)
END-READ
END

```

Ausgabe des Programms PERFBX01:

Page 1 04-12-14 14:13:35

| NAME                              | DEPARTMENT<br>CODE | ANNUAL<br>SALARY |         |                |
|-----------------------------------|--------------------|------------------|---------|----------------|
| JENSEN                            | ADMA01             | 180000           |         |                |
| PETERSEN                          | ADMA01             | 105000           |         |                |
| MORTENSEN                         | ADMA01             | 320000           |         |                |
| MADSEN                            | ADMA01             | 149000           |         |                |
| BUHL                              | ADMA01             | 642000           |         |                |
| SUMMARY FOR ALL SALARIES          |                    | SUM:             | 1396000 | TOTAL: 1396000 |
| SUMMARY FOR SALARY GREATER 100000 |                    | SUM:             | 1396000 | TOTAL: 1396000 |

|                                   |        |         |        |         |
|-----------------------------------|--------|---------|--------|---------|
| SUMMARY FOR SALARY GREATER 150000 | SUM:   | 1142000 | TOTAL: | 1142000 |
| HERMANSEN                         | ADMA02 | 391500  |        |         |
| PLOUG                             | ADMA02 | 162900  |        |         |

|                                   |      |        |        |         |
|-----------------------------------|------|--------|--------|---------|
| SUMMARY FOR ALL SALARIES          | SUM: | 554400 | TOTAL: | 1950400 |
| SUMMARY FOR SALARY GREATER 100000 | SUM: | 554400 | TOTAL: | 1950400 |
| SUMMARY FOR SALARY GREATER 150000 | SUM: | 554400 | TOTAL: | 1696400 |