

AT END OF PAGE

Structured Mode-Syntax

```
[AT] END [OF] PAGE [(rep)]
    statement ...
END-ENDPAGE
```

Reporting Mode-Syntax

```
[AT] END [OF] PAGE [(rep)]
{
    statement
    DO statement ... DOEND
}
```

Dieses Kapitel behandelt folgende Themen:

- Funktion
- Syntax-Beschreibung
- Beispiele

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

Verwandte Statements: AT TOP OF PAGE | CLOSE PRINTER | DEFINE PRINTER | DISPLAY | EJECT | FORMAT | NEWPAGE | PRINT | SKIP | SUSPEND IDENTICAL SUPPRESS | WRITE | WRITE TITLE | WRITE TRAILER

Gehört zur Funktionsgruppe: *Erstellen von Ausgabe-Reports*

Funktion

Mit dem AT END OF PAGE-Statement können Sie eine Verarbeitung angeben, die ausgeführt werden soll, wenn das Ende einer logischen Seite erreicht ist (End-of-Page-Bedingung; siehe Session-Parameter PS in der *Parameter-Referenz*). Eine End-of-Page-Bedingung kann auch aufgrund eines SKIP- oder NEWPAGE-Statements auftreten, nicht aber aufgrund eines EJECT- oder INPUT-Statements.

Siehe auch die folgenden Abschnitte im *Leitfaden zur Programmierung*:

- *Steuerung der Ausgabe von Daten*
- *Report-Spezifikation — (rep)-Notation*

- *Layout einer Ausgabeseite*
- *AT END OF PAGE-Statement*

Verarbeitung

Ein `AT END OF PAGE`-Statement-Block wird nur ausgeführt, wenn das Objekt, das den Statement-Block enthält, zu dem Zeitpunkt, zu dem die End-of-Page-Bedingung auftritt, aktiv ist.

Ein `AT END OF PAGE`-Statement darf nicht in einer internen Subroutine stehen.

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

Länge der logischen Seite

Da erst überprüft wird, ob eine End-of-Page-Bedingung besteht, nachdem ein `DISPLAY`- oder `WRITE`-Statement vollständig ausgeführt ist, kann es vorkommen, dass die von einem `DISPLAY`- oder `WRITE`-Statement erzeugte Ausgabe das Ende einer physischen Seite bereits überschritten hat, bevor eine End-of-Page-Bedingung entdeckt wird.

Um dies zu vermeiden und um sicherzustellen, dass über ein `AT END OF PAGE`-Statement ausgegebene Informationen wirklich am Ende einer physischen Ausgabeseite erscheint, muss die logische Seitenlänge (Session-Parameter `PS`) entsprechend kleiner als die Länge einer physischen Ausgabeseite gesetzt werden.

Letzte Seite

In einem Hauptprogramm ist eine End-of-Page-Bedingung auch dann gegeben, wenn die Ausführung des Programms durch ein `ESCAPE`-, `STOP`- oder `END`-Statement beendet wird.

In einer Subroutine gilt dies nicht; das heißt, `ESCAPE-ROUTINE`, `RETURN` oder `END-SUBROUTINE` lösen in einer Subroutine keine End-of-Page-Bedingung aus.

Systemfunktionen

Natural-Systemfunktionen können in Verbindung mit einem `AT END OF DATA`-Statement verwendet werden, wie im Abschnitt *Systemfunktionen für Verarbeitungsschleifen* benutzen in der *Systemfunktionen*-Dokumentation beschrieben.

Wenn eine Systemfunktion in einem `AT END OF PAGE`-Statement-Block verwendet wird, muss das betreffende `DISPLAY`-Statement eine `GIVE SYSTEM FUNCTIONS`-Klausel enthalten.

INPUT-Statement im AT END OF PAGE

Wenn Sie im `AT END OF PAGE`-Block ein `INPUT`-Statement verwenden, wird keine Seitenvorschub-Operation ausgeführt. Sie müssen in diesem Fall den Wert des Session-Parameters `PS` soweit reduzieren, dass die vom `INPUT`-Statement erzeugten Zeilen noch auf derselben physischen Seite Platz haben.

Siehe auch:

- *Geteilter Schirm (Split Screen)* beim INPUT-Statement
- *Beispiel 2 — AT END OF PAGE mit INPUT-Statement*

Syntax-Beschreibung

<i>(rep)</i>	<p>Report-Spezifikation: Mit der Notation (<i>rep</i>) kann ein bestimmter anderer Report angegeben werden, auf den sich das Statement beziehen soll. Es kann ein Wert von 0 bis 31 oder ein logischer Name, der mit einem DEFINE PRINTER-Statement zugewiesen wurde, angegeben werden.</p> <p>Falls (<i>rep</i>) nicht angegeben wird, bezieht sich das AT END OF PAGE-Statement auf den ersten Report (Report 0).</p> <p>Informationen zum Steuern des Formats eines mit Natural erzeugten Ausgabe-Reports siehe <i>Steuerung der Ausgabe von Daten im Leitfaden zur Programmierung</i>.</p>
END-ENDPAGE	Das reservierte Natural-Wort END-ENDPAGE muss zum Beenden des AT END OF PAGE-Statements benutzt werden.

Beispiele

- Beispiel 1 — AT END OF PAGE
- Beispiel 2 — AT END OF PAGE mit INPUT-Statement

Beispiel 1 — AT END OF PAGE

```

** Example 'AEPEx1S': AT END OF PAGE (structured mode)
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 PERSONNEL-ID
  2 NAME
  2 JOB-TITLE
  2 SALARY      (1)
  2 CURR-CODE  (1)
END-DEFINE
*
FORMAT PS=10
LIMIT 10
READ EMPLOY-VIEW BY PERSONNEL-ID FROM '20017000'
  DISPLAY NOTITLE GIVE SYSTEM Funktions
      NAME JOB-TITLE 'SALARY' SALARY(1) CURR-CODE (1)
/*

AT END OF PAGE
  WRITE / 28T 'AVERAGE SALARY: ...' AVER(SALARY(1)) CURR-CODE (1)
END-ENDPAGE

```

```
END-READ
*
END
```

Siehe auch *Systemfunktionen für Verarbeitungsschleifen*.

Ausgabe des Programms AEPEX1S:

NAME	CURRENT POSITION	SALARY	CURRENCY CODE
-----	-----	-----	-----
CREMER	ANALYST	34000	USD
MARKUSH	TRAINEE	22000	USD
GEE	MANAGER	39500	USD
KUNEY	DBA	40200	USD
NEEDHAM	PROGRAMMER	32500	USD
JACKSON	PROGRAMMER	33000	USD
	AVERAGE SALARY: ...	33533	USD

Äquivalentes Reporting-Mode-Beispiel: AEPEX1R.

Beispiel 2 — AT END OF PAGE mit INPUT-Statement

```
** Example 'AEPEX2': AT END OF PAGE (with INPUT)
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 NAME
  2 FIRST-NAME
  2 POST-CODE
  2 CITY
*
1 #START-NAME (A20)
END-DEFINE
*
FORMAT PS=21
*
REPEAT
  READ (15) EMPLOY-VIEW BY NAME = #START-NAME
  DISPLAY NOTITLE NAME FIRST-NAME POST-CODE CITY
  END-READ
  NEWPAGE
  /*
  AT END OF PAGE
  MOVE NAME TO #START-NAME
  INPUT / '-' (79)
    / 10T 'Reposition to name ==>'
      #START-NAME (AD=MI) '('''.' to exit)'
  IF #START-NAME = '.'
    STOP
  END-IF
  END-ENDPAGE
  /*
END-REPEAT
END
```

Ausgabe des Programms AEPX2S:

NAME	FIRST-NAME	POSTAL ADDRESS	CITY
ABELLAN	KEPA	28014	MADRID
ACHIESON	ROBERT	DE3 4TR	DERBY
ADAM	SIMONE	89300	JOIGNY
ADKINSON	JEFF	11201	BROOKLYN
ADKINSON	PHYLLIS	90211	BEVERLEY HILLS
ADKINSON	HAZEL	20760	GAITHERSBURG
ADKINSON	DAVID	27514	CHAPEL HILL
ADKINSON	CHARLIE	21730	LEXINGTON
ADKINSON	MARTHA	17010	FRAMINGHAM
ADKINSON	TIMMIE	17300	BEDFORD
ADKINSON	BOB	66044	LAWRENCE
AECKERLE	SUSANNE	7000	STUTTGART
AFANASSIEV	PHILIP	39401	HATTIESBURG
AFANASSIEV	ROSE	60201	EVANSTON
AHL	FLEMMING	2300	SUNDBY

Reposition to name ==> AHL ('.' to exit)