

PERFORM BREAK PROCESSING

```
PERFORM BREAK [PROCESSING] [(r)]
```

```
AT BREAK statement ...
```

This chapter covers the following topics:

- Function
- Syntax Description
- Example

For an explanation of the symbols used in the syntax diagram, see *Syntax Symbols*.

Related Statements: ACCEPT/REJECT | AT BREAK | 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 | READ | RETRY | STORE | UPDATE

Belongs to Function Group: *Database Access and Update*

Function

The PERFORM BREAK PROCESSING statement is used to establish break processing in loops created by FOR, REPEAT, CALL LOOP and CALL FILE statements where no automatic break processing is established, or whenever a user-initiated break processing is desired. Unlike automatic break processing which is executed immediately after the record is read, the PERFORM BREAK PROCESSING statement is executed when it is encountered in the normal flow of the program.

This statement causes a check for a break processing condition (based on the value of a control field) and also results in the evaluation of Natural system functions. This check and system function evaluation are performed each time the statement is encountered for execution. This statement may be executed depending on a condition specified in an IF statement.

Syntax Description

<p>(r)</p>	<p>Statement Reference Notation:</p> <p>By default, the final PERFORM BREAK condition is true at the end of execution of the program, subprogram or subroutine.</p> <p>The notation (r) may be used to relate the final processing of a PERFORM BREAK to a specific loop. In this case the PERFORM BREAK is executed in the loop end handling of this loop; after the final automatic BREAK processing and before the AT END OF DATA statements are executed.</p>
<p>AT BREAK <i>statement...</i></p>	<p>See the syntax of the AT BREAK statement.</p>

Example

```

** Example 'PBPEX1S': PERFORM BREAK PROCESSING (structured mode)
*****
DEFINE DATA LOCAL
1 #INDEX (N2)
1 #LINE (N2) INIT <1>
END-DEFINE
*
FOR #INDEX 1 TO 18
  PERFORM BREAK PROCESSING
  /*
  AT BREAK OF #INDEX /1/
  WRITE NOTITLE / 'PLEASE COMPLETE LINES 1-9 ABOVE' /
  RESET INITIAL #LINE
  END-BREAK
  /*
  WRITE NOTITLE '_' (64) '=' #LINE
  ADD 1 TO #LINE
END-FOR
*
END
    
```

Output of Program PBPEX1S:

```

_____ #LINE: 1
_____ #LINE: 2
_____ #LINE: 3
_____ #LINE: 4
_____ #LINE: 5
_____ #LINE: 6
_____ #LINE: 7
_____ #LINE: 8
_____ #LINE: 9
    
```

PLEASE COMPLETE LINES 1-9 ABOVE

```

_____ #LINE: 1
_____ #LINE: 2
_____ #LINE: 3
_____ #LINE: 4
_____ #LINE: 5
_____ #LINE: 6
_____ #LINE: 7
    
```

#LINE: 8

#LINE: 9

PLEASE COMPLETE LINES 1-9 ABOVE

Equivalent reporting-mode example: PBPEX1R.