REPEAT

This chapter covers the following topics:

- Function

- Syntax Description

- Examples

Related Statements: FOR | ESCAPE

Belongs to Function Group: *Loop Execution*

Function

The REPEAT statement is used to initiate a processing loop.

Syntax Description
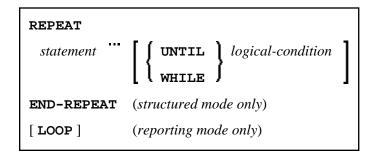
Two different structures are possible for this statement.

- Syntax 1 - Statements are executed one or more times

- Syntax 2 - Statements are executed zero or more times

The placement of the logical condition (either at the beginning or at the end of the loop) determines when it is to be evaluated.

For further information on logical conditions, see the section *Logical Condition Criteria* in the *Programming Guide*.

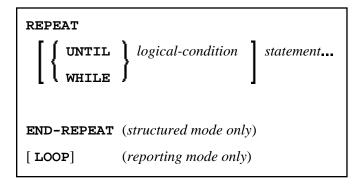For an explanation of the symbols used in the syntax diagrams, see *Syntax Symbols*.

Syntax 1:

```
REPEAT

  statement  ···  [ { UNTIL  } logical-condition ]
                  { WHILE  }

END-REPEAT    (structured mode only)
[ LOOP ]        (reporting mode only)
```

Syntax 2:

```
REPEAT

  [ { UNTIL }  logical-condition  ]  statement...
  [ { WHILE }                     ]


END-REPEAT  (structured mode only)

[ LOOP]         (reporting mode only)
```

Syntax Element Description:

Syntax Element	Description
UNTIL	**UNTIL Option:** The processing loop will be continued until the logical condition becomes true.
WHILE	**WHILE Option:** The processing loop will be continued as long as the logical condition is true.
logical-condition	**Logical Condition:** If a logical condition is specified, the condition determines when the execution of the loop is to be terminated. If no logical condition is specified, the loop must be exited by an ESCAPE, STOP or TERMINATE statement specified within the loop. The syntax for a logical condition is described in the section *Logical Condition Criteria* in the Programming Guide.
END-REPEAT	**End of REPEAT Statement:** The Natural reserved word END-REPEAT must be used to end the REPEAT statement.

Examples

- Example 1 - REPEAT

- Example 2 - Using WHILE and UNTIL Options

Example 1 - REPEAT

```
** Example 'RPTEX1S': REPEAT (structured mode)
************************************************************************
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 PERSONNEL-ID
  2 NAME
```

Example 2 - Using WHILE and UNTIL Options   **REPEAT**

```
*
1 #PERS-NR (A8)
END-DEFINE
*
REPEAT
  INPUT 'ENTER A PERSONNEL NUMBER:' #PERS-NR
  IF #PERS-NR = ' '
    ESCAPE BOTTOM
  END-IF
  /*
  FIND EMPLOY-VIEW WITH PERSONNEL-ID = #PERS-NR
    IF NO RECORD FOUND
      REINPUT 'NO RECORD FOUND'
    END-NOREC
    DISPLAY NOTITLE NAME
  END-FIND
END-REPEAT
*
END
```

Output of Program RPTEX1S:

```
ENTER A PERSONNEL NUMBER: 11500304
```

After entering and confirming personnel number:

```
        NAME
-------------------

KLUGE
```

Equivalent reporting-mode example: RPTEX1R.

Example 2 - Using WHILE and UNTIL Options

```
** Example 'RPTEX2S': REPEAT (with WHILE and UNTIL option)
************************************************************************
DEFINE DATA LOCAL
1 #X (I1) INIT <0>
1 #Y (I1) INIT <0>
END-DEFINE
*
REPEAT WHILE #X <= 5
  ADD 1 TO #X
  WRITE NOTITLE '=' #X
END-REPEAT
*
SKIP 3
REPEAT
  ADD 1 TO #Y
  WRITE '=' #Y
  UNTIL #Y = 6
END-REPEAT
*
END
```

Output of Program RPTEX2S:

```
#X:     1
#X:     2
#X:     3
#X:     4
#X:     5
#X:     6


#Y:     1
#Y:     2
#Y:     3
#Y:     4
#Y:     5
#Y:     6
```

Equivalent reporting-mode example: RPTEX2R.