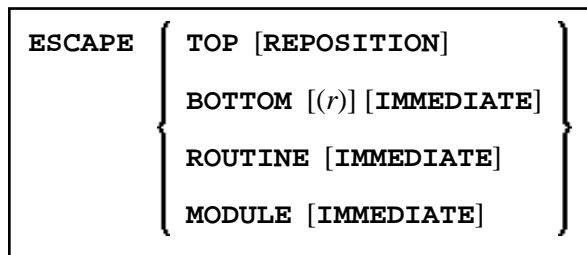
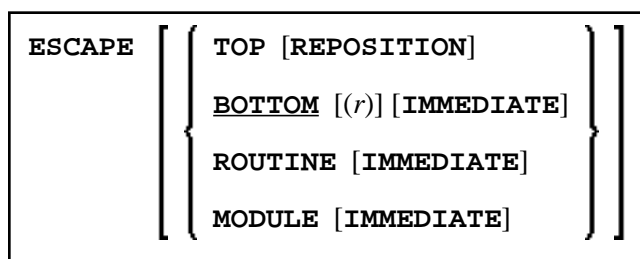


# ESCAPE

## Structured Mode Syntax



## Reporting Mode Syntax



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:

- FOR | REPEAT | PROCESS PAGE MODAL
- CALL | CALL FILE | CALL LOOP | CALLNAT | DEFINE SUBROUTINE | FETCH | PERFORM

Belongs to Function Group:

- *Loop Execution*
- *Invoking Programs and Routines*

## Function

The ESCAPE statement is used to interrupt the linear flow of execution of a processing loop or a routine.

With the keywords TOP, BOTTOM and ROUTINE you indicate where processing is to continue when the ESCAPE statement is encountered.

An ESCAPE TOP/BOTTOM statement, when encountered for processing, will internally refer to the innermost active processing loop. The ESCAPE statement need not be physically placed within the processing loop.

If an ESCAPE TOP/BOTTOM statement is placed in a routine (subroutine, subprogram, or program invoked with FETCH RETURN) , the routine(s) entered during execution of the processing loop will be terminated automatically.

## Additional Considerations

More than one ESCAPE statement may be contained within the same processing loop.

The execution of an ESCAPE statement may be based on a logical condition. If an ESCAPE statement is encountered during processing of an AT END OF DATA, AT BREAK or AT END OF PAGE block, the execution of the special condition block will be terminated and ESCAPE processing will continue as required.

If an ESCAPE statement is encountered during processing of an if-no-records-found condition, no loop-end processing will be performed (equivalent to ESCAPE IMMEDIATE).

## Syntax Description

Syntax Element	Description
ESCAPE TOP	<p><b>Top Option:</b></p> <p>TOP indicates that processing is to continue at the top of the processing loop. This starts the next repetition of the processing loop.</p>
REPOSITION	<p><b>Top Reposition Option:</b></p> <p>When an ESCAPE TOP REPOSITION statement is executed, Natural immediately continues processing at the top of the active READ loop, using the current value of the search variable as new start value.</p> <p>At the same time, ESCAPE TOP REPOSITION resets the system variable *COUNTER to zero.</p> <p>ESCAPE TOP REPOSITION can be specified within a READ statement loop accessing an Adabas database. The READ statement concerned must contain the option WITH REPOSITION.</p>
ESCAPE BOTTOM	<p><b>Bottom Option:</b></p> <p>BOTTOM indicates that processing is to continue with the first statement following the processing loop. The loop is terminated and loop-end processing (final BREAK and END DATA) is executed for all loops being terminated.</p> <p>In reporting mode, ESCAPE BOTTOM is the default.</p>

Syntax Element	Description
( <i>r</i> )	<p><b>Statement Reference:</b></p> <p>Notation ( <i>r</i> ): If BOTTOM is followed by a label or reference number, processing will continue with the first statement following the processing loop identified by the label or reference number.</p> <p>A label or a reference number can only be specified if the ESCAPE BOTTOM statement is physically placed within the referenced processing loop.</p>
IMMEDIATE	<p><b>Immediate Option:</b></p> <p>If you specify the keyword IMMEDIATE, no loop-end processing will be performed.</p>
ESCAPE ROUTINE	<p><b>Routine Option:</b></p> <p>This option indicates that the current Natural routine, which may have been invoked via a PERFORM, CALLNAT, FETCH RETURN, or as a main program, is to relinquish control.</p> <p>In the case of a subroutine, processing will continue with the first statement after the statement used to invoke the subroutine. In the case of a main program, Natural command mode will be entered.</p> <p>All loops currently active within the routine will be terminated and loop-end processing performed as well as final processing for user-initiated (PERFORM BREAK) processing. If the program containing the ESCAPE ROUTINE is executed as a main program (Level 1), final end-page processing is performed.</p>
ESCAPE MODULE	<p><b>Module Option:</b></p> <p>This option indicates that the entire current program level, with all internal subroutines, is to relinquish control. The control is then returned to the object of the former program level. If ESCAPE MODULE is used in a hierarchy of internal subroutines, it allows to escape all routines working at this level at once. If no internal subroutine is active, ESCAPE MODULE has the same result as ESCAPE ROUTINE.</p> <p>ESCAPE MODULE is only relevant in inline subroutines. In external subroutines, subprograms and invoked programs, it has the same effect as ESCAPE ROUTINE.</p> <p>As with ESCAPE ROUTINE, loop-end processing will be performed. However, if you specify the keyword IMMEDIATE, no loop-end processing will be performed.</p>

## Example

```

** Example 'ESCEX1': ESCAPE
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES

```

```

2 CITY
2 FIRST-NAME
2 NAME
2 AREA-CODE
2 PHONE
*
1 #CITY (A20) INIT <' '>
1 #CNTL (A1) INIT <' '>
END-DEFINE
*
REPEAT
  INPUT 'ENTER VALUE FOR CITY: ' #CITY
    / 'OR ''.' TO TERMINATE '
  IF #CITY = '.'
    ESCAPE BOTTOM
  END-IF
/*
FND. FIND EMPLOY-VIEW WITH CITY = #CITY
/*
  IF NO RECORDS FOUND
    WRITE 'NO RECORDS FOUND'
    ESCAPE BOTTOM (FND.)
  END-NOREC
  AT START OF DATA
    INPUT (AD=0) 'RECORDS FOUND:' *NUMBER //
      'ENTER ''D'' TO DISPLAY RECORDS' #CNTL (AD=M)
    IF #CNTL NE 'D'
      ESCAPE BOTTOM (FND.)
    END-IF
  END-START
/*
  DISPLAY NOTITLE NAME FIRST-NAME PHONE
END-FIND
END-REPEAT

```

### Output of Program ESCEX1:

```

ENTER VALUE FOR CITY:  PARIS
(OR '.' TO TERMINATE)

```

### After entering and confirming city name:

```

RECORDS FOUND:          26
ENTER 'D' TO DISPLAY RECORDS D

```

### Result after entering and confirming D:

NAME	FIRST-NAME	TELEPHONE
MAIZIERE	ELISABETH	46758304
MARX	JEAN-MARIE	40738871
REIGNARD	JACQUELINE	48472153
RENAUD	MICHEL	46055008
REMOUE	GERMAINE	36929371
LAVENDA	SALOMON	40155905
BROUSSE	GUY	37502323
GIORDA	LOUIS	37497316
SIECA	FRANCOIS	40487413
CENSIER	BERNARD	38070268
DUC	JEAN-PAUL	38065261
CAHN	RAYMOND	43723961

MAZUY	ROBERT	44286899
FAURIE	HENRI	44341159
VALLY	ALAIN	47326249
BRETON	JEAN-MARIE	48467146
GIGLEUX	JACQUES	40477399
KORAB-BRZOWSKI	BOGDAN	45288048
XOLIN	CHRISTIAN	46060015
LEGRIS	ROGER	39341509
VVVV		