

ON ERROR

Structured Mode Syntax

```
ON ERROR
  statement...
END-ERROR
```

Reporting Mode Syntax

```
ON ERROR { statement
           DO statement...DOEND }
```

Function

The ON ERROR statement is used to intercept execution time errors which would otherwise result in a Natural error message, followed by termination of Natural program execution, and a return to command input mode.

When the ON ERROR statement block is entered for execution, the normal flow of program execution has been interrupted and cannot be resumed except for error 3145 (record requested in hold), in which case a RETRY statement will cause processing to be resumed exactly where it was suspended.

This statement is non-procedural (that is, its execution depends on an event, not on where in a program it is located).

ON ERROR Processing within Subroutines

When a subroutine structure is built by using CALLNAT, PERFORM or FETCH RETURN, each module may contain an ON ERROR statement.

When an error occurs, Natural will automatically trace back the subroutine structure and select the first ON ERROR statement encountered in a subroutine for execution. If no ON ERROR statement is found in any module on any level, standard error message processing is performed and program execution is terminated.

Restriction

Only one ON ERROR statement is permitted in a Natural object.

System Variables *ERROR-NR and *ERROR-LINE

The Natural system variable *ERROR-NR contains the number of the error detected by Natural.

The Natural system variable *ERROR-LINE contains the line number of the statement which caused the error.

Exiting from an ON ERROR Block

An ON ERROR block may be exited by using a FETCH, STOP, TERMINATE, RETRY or ESCAPE ROUTINE statement. If the block is not exited using one of these statements, standard error message processing is performed and program execution is terminated.

Example

```

/* EXAMPLE 'ONEEX1': ON ERROR
/*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 NAME
  2 CITY
1 #NAME (A20)
1 #CITY (A20)
END-DEFINE
/*****
REPEAT
INPUT 'ENTER NAME:' #NAME
IF #NAME = ' '
  STOP
END-IF
FIND EMPLOY-VIEW WITH NAME = #NAME
INPUT (AD=M) 'ENTER NEW VALUES:' ///
      'NAME:' NAME /
      'CITY:' CITY

UPDATE
END TRANSACTION
/*****
ON ERROR
IF *ERROR-NR = 3009
  WRITE 'LAST TRANSACTION NOT SUCCESSFUL'
  / 'HIT ENTER TO RESTART PROGRAM'
  FETCH 'PROGUPD'
END-IF
END-ERROR
/*****
END-FIND
END-REPEAT
END

```