

# BACKOUT TRANSACTION

**BACKOUT [TRANSACTION]**

This chapter covers the following topics:

- Function
- Restriction
- Database-Specific Considerations
- 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 | BEFORE BREAK PROCESSING | DELETE | END TRANSACTION | FIND | GET | GET SAME | GET TRANSACTION DATA | HISTOGRAM | LIMIT | PASSW | PERFORM BREAK PROCESSING | READ | RETRY | STORE | UPDATE

Belongs to Function Group: *Database Access and Update*

---

## Function

The `BACKOUT TRANSACTION` statement is used to back out all database updates performed during the current logical transaction. This statement also releases all records held during the transaction.

The statement is executed only if a database transaction under control of Natural has taken place. For which databases the statement is executed depends on the setting of the profile parameter `ET` (execution of `END/BACKOUT TRANSACTION` statements):

- If `ET=OFF`, the statement is executed only for the database affected by the transaction.
- If `ET=ON`, the statement is executed for all databases that have been referenced since the last execution of a `BACKOUT TRANSACTION` or `END TRANSACTION` statement.

## Backout Transaction Issued by Natural

If the user interrupts the current Natural operation with a terminal command (command `%%` or `CLEAR` key), Natural issues a `BACKOUT TRANSACTION` statement.

See also the terminal command `%%` in the *Terminal Commands* documentation.

## Additional Information

For additional information on the use of the transaction backout feature, see the sections *Database Update - Transaction Processing* and *Backing Out a Transaction* in the *Programming Guide*.

## Restriction

This statement is not available with Entire System Server.

## Database-Specific Considerations

<b>SQL Databases</b>	As most SQL databases close all cursors when a logical unit of work ends, a BACKOUT TRANSACTION statement must not be placed within a database modification loop; instead, it has to be placed after such a loop.
<b>XML Databases</b>	A BACKOUT TRANSACTION statement must not be placed within a database modification loop; instead, it has to be placed after such a loop.

## Example

```

** Example 'BOTEX1': BACKOUT TRANSACTION
**
** CAUTION: Executing this example will modify the database records!
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 NAME
  2 DEPT
  2 LEAVE-DUE
  2 LEAVE-TAKEN
*
1 #DEPT (A6)
1 #RESP (A3)
END-DEFINE
*
LIMIT 3
INPUT 'DEPARTMENT TO BE UPDATED:' #DEPT
IF #DEPT = ' '
  STOP
END-IF
*
FIND EMPLOY-VIEW WITH DEPT = #DEPT
  IF NO RECORDS FOUND
    REINPUT 'NO RECORDS FOUND'
  END-NOREC
  INPUT 'NAME:          ' NAME (AD=O) /
        'LEAVE DUE:    ' LEAVE-DUE (AD=M) /
        'LEAVE TAKEN:' LEAVE-TAKEN (AD=M)

  UPDATE
END-FIND
*
INPUT 'UPDATE TO BE PERFORMED? YES/NO:' #RESP
DECIDE ON FIRST #RESP
  VALUE 'YES'
  END TRANSACTION
  VALUE 'NO'
  BACKOUT TRANSACTION
  NONE

```

```
REINPUT 'PLEASE ENTER YES OR NO'  
END-DECIDE  
*  
END
```

### Output of Program BOTEX1:

DEPARTMENT TO BE UPDATED: **MGMT30**

Result for department MGMT30:

```
NAME:          POREE  
LEAVE DUE:     45  
LEAVE TAKEN:  31
```

Confirmation query:

UPDATE TO BE PERFORMED YES/NO: **NO**