

GET TRANSACTION DATA

`GET TRANSACTION [DATA] operand1 ...`

This chapter covers the following topics:

- Function
- Restriction
- 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 | HISTOGRAM | LIMIT | PASSW | PERFORM BREAK PROCESSING | READ | RETRY | STORE | UPDATE

Belongs to Function Group: *Database Access and Update*

Function

The `GET TRANSACTION DATA` statement is used to read the data saved with a previous `END TRANSACTION` statement.

`GET TRANSACTION DATA` does not create a processing loop.

Note:

For DL/I: The `GET TRANSACTION DATA` statement retrieves checkpoint data saved by an `END TRANSACTION` statement.

System Variable *ETID

The content of the Natural system variable *ETID identifies the transaction data to be retrieved from the database.

No Transaction Data Stored

If the `GET TRANSACTION DATA` statement is issued and no transaction data are found, all fields specified in the `GET TRANSACTION DATA` statement will be filled with blanks regardless of format definition.



Warning:

Make sure that arithmetic operations are not performed on "empty" transaction data, because this would result in an abnormal termination of the program.

Restriction

The GET TRANSACTION DATA statement is only valid for transactions applied to Adabas databases, or to DL/I databases in a batch-oriented BMP region (in IMS TM environments only).

Syntax Description

Operand Definition Table:

Operand	Possible Structure	Possible Formats	Referencing Permitted	Dynamic Definition
<i>operand1</i>	S	A U N P I F B D T	yes	yes

Syntax Element Description:

<i>operand1</i>	<p>Field Specification:</p> <p>The sequence, lengths and formats of the fields used in the GET TRANSACTION DATA statement must be identical to the sequence, lengths and formats of the fields specified with the corresponding END TRANSACTION statement.</p> <p>Note: For DL/I: The first <i>operand1</i> must be an 8-byte checkpoint ID.</p>
-----------------	--

Example

```

** Example 'GTREX1': GET TRANSACTION
**
** CAUTION: Executing this example will modify the database records!
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 PERSONNEL-ID
  2 NAME
  2 FIRST-NAME
  2 MIDDLE-I
  2 CITY
*
1 #PERS-NR (A8) INIT <' '>
END-DEFINE
*
GET TRANSACTION DATA #PERS-NR
IF #PERS-NR NE ' '
  WRITE 'LAST TRANSACTION PROCESSED FROM PREVIOUS SESSION' #PERS-NR
END-IF
*
REPEAT
/*
  INPUT 10X 'ENTER PERSONNEL NUMBER TO BE UPDATED:' #PERS-NR
  IF #PERS-NR = ' '
    STOP
  END-IF
/*

```

```
FIND EMPLOY-VIEW WITH PERSONNEL-ID = #PERS-NR
  IF NO RECORDS FOUND
    REINPUT 'NO RECORD FOUND'
  END-NOREC
  INPUT (AD=M) PERSONNEL-ID (AD=O)
    / NAME
    / FIRST-NAME
    / CITY

  UPDATE
  END TRANSACTION #PERS-NR
END-FIND
/*
END-REPEAT
END
```