

# GET

In structured mode and in reporting mode using a `DEFINE DATA LOCAL` statement, the following syntax applies:

```

GET [IN] [FILE] view-name
    [PASSWORD=operand1]
    [CIPHER=operand2]
    [ { [RECORD] } ] { operand3 }
    [ { [RECORDS] } ] { *ISN [(r)] }

```

In reporting mode using no `DEFINE DATA LOCAL` statement, the following syntax applies:

```

GET [IN] [FILE] ddm-name
    [PASSWORD=operand1]
    [CIPHER=operand2]
    [ { [RECORD] } ] { operand3 } operand4 ...
    [ { [RECORDS] } ] { *ISN [(r)] }

```

This chapter covers the following topics:

- Function
- Restrictions
- 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 SAME` | `GET TRANSACTION` | `HISTOGRAM` | `LIMIT` | `PASSW` | `PERFORM BREAK PROCESSING` | `READ` | `RETRY` | `STORE` | `UPDATE`

Belongs to Function Group: *Database Access and Update*

---

## Function

The `GET` statement is used to read a record with a given Adabas Internal Sequence Number (ISN).

For XML databases, the GET statement is used to read an XML object with a given object ID.

The GET statement does not cause a processing loop to be initiated.

## Restrictions

- The GET statement cannot be used for SQL databases.
- The GET statement cannot be used with Entire System Server.

## Syntax Description

Operand Definition Table:

Operand	Possible Structure		Possible Formats												Referencing Permitted	Dynamic Definition			
<i>operand1</i>	C	S				A												yes	no
<i>operand2</i>	C	S					N											no	no
<i>operand3</i>	C	S			N		N	P	I		B							yes	no
<i>operand4</i>		S	A			A	N	P	I	F	B	D	T	L				yes	yes

\* Format B of *operand3* may be used only with a length of less than or equal to 4.

Syntax Element Description:

Syntax Element	Description
<i>view-name</i>	<p><b>View Name:</b></p> <p>In structured mode and in reporting mode using a DEFINE DATA LOCAL statement, the name of a view as defined either directly within a DEFINE DATA statement or in a separate global or local data area.</p>
<i>dgm-name</i>	<p><b>DDM Name:</b></p> <p>In reporting mode using no DEFINE DATA LOCAL statement, the name of the data definition module (DDM) is referenced.</p>
PASSWORD= <i>operand1</i>	<p><b>PASSWORD Clause/CYPHER Clause:</b></p> <p>These clauses are applicable only to Adabas databases.</p> <p>The PASSWORD clause is used to provide a password when retrieving data from an Adabas file which is password protected.</p> <p>The CIPHER clause is used to provide a cipher key when retrieving data from an Adabas file which is enciphered.</p> <p>See the statements FIND and PASSW for further information.</p>
CIPHER= <i>operand2</i>	
*ISN / <i>operand3</i>	<p><b>Internal Sequence Number:</b></p> <p>The ISN must be provided either in the form of a numeric constant or user-defined variable (<i>operand3</i>), or via the Natural system variable *ISN.</p>
( <i>r</i> )	<p><b>Statement Reference:</b></p> <p>The notation (<i>r</i>) is used to specify the statement which contains the FIND or READ statement used to initially read the record.</p> <p>If (<i>r</i>) is not specified, the GET statement will be related to the innermost active processing loop.</p> <p>(<i>r</i>) may be specified as a reference statement number or as a statement label.</p>
<i>operand4</i>	<p><b>Reference to Database Fields:</b></p> <p>In reporting mode, subsequent references to database fields that have been read with a GET statement can contain the label or line number of the GET statement.</p>

## Example

```

** Example 'GETEX1': GET
*****
DEFINE DATA LOCAL
1 PERSONS VIEW OF EMPLOYEES
  2 PERSONNEL-ID
  2 NAME

```

```

      2 FIRST-NAME
1 SALARY-INFO VIEW OF EMPLOYEES
      2 NAME
      2 CURR-CODE (1:1)
      2 SALARY    (1:1)
*
1 #ISN-ARRAY  (B4/1:10)
1 #LINE-NR    (N2)
END-DEFINE
*
FORMAT PS=16
LIMIT 10
READ PERSONS BY NAME
      MOVE *COUNTER TO #LINE-NR
      MOVE *ISN     TO #ISN-ARRAY (#LINE-NR)
      DISPLAY #LINE-NR PERSONNEL-ID NAME FIRST-NAME
/*
AT END OF PAGE
      INPUT / 'PLEASE SELECT LINE-NR FOR SALARY INFORMATION:' #LINE-NR
      IF #LINE-NR = 1 THRU 10
          GET SALARY-INFO #ISN-ARRAY (#LINE-NR)
          WRITE / SALARY-INFO.NAME
                  SALARY-INFO.SALARY (1)
                  SALARY-INFO.CURR-CODE (1)

          END-IF
      END-ENDPAGE
/*
END-READ
END

```

### Output of Program GETEX1:

Page 1

05-01-13 13:17:42

#LINE-NR	PERSONNEL ID	NAME	FIRST-NAME
----------	-----------------	------	------------

1	60008339	ABELLAN	KEPA
2	30000231	ACHIESON	ROBERT
3	50005800	ADAM	SIMONE
4	20008800	ADKINSON	JEFF
5	20009800	ADKINSON	PHYLLIS
6	20012700	ADKINSON	HAZEL
7	20013800	ADKINSON	DAVID
8	20019600	ADKINSON	CHARLIE
9	20008600	ADKINSON	MARTHA
10	20005700	ADKINSON	TIMMIE

PLEASE SELECT LINE-NR FOR SALARY INFORMATION: 1

ABELLAN 1450000 PTA