

STORE

Structured Mode Syntax

```

STORE [RECORD] [IN] [FILE] view-name
      [PASSWORD=operand1]
      [CIPHER=operand2]
      [ [ USING                ] NUMBER operand3 ]
      [ [ GIVING                ]                ]

```

Reporting Mode Syntax

```

STORE [RECORD] [IN] [FILE] view-name
      [PASSWORD=operand1]
      [CIPHER=operand2]
      [ [ USING                ] NUMBER operand3 ]
      [ [ GIVING                ]                ]
      { [ [ USING] SAME [RECORD] [AS] [STATEMENT [(r)] ]
        [ [ SET                ] [operand4=operand5] ... ]
        [ [ WITH                ]                ]                }

```

This chapter covers the following topics:

- Function
- Database-Specific Considerations
- 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 | GET TRANSACTION DATA | HISTOGRAM | LIMIT | PASSW | PERFORM BREAK PROCESSING | READ | RETRY | UPDATE

Belongs to Function Group: *Database Access and Update*

Syntax Element	Description
<i>view-name</i>	<p>View Name:</p> <p>As <i>view-name</i>, you specify the name of a view, which must have been defined either in a DEFINE DATA statement or outside the program in a global or local data area.</p> <p>In reporting mode, <i>view-name</i> is the name of a DDM if no DEFINE DATA LOCAL statement is used.</p>
PASSWORD= <i>operand1</i>	<p>PASSWORD Clause:</p> <p>The PASSWORD clause is applicable only for an Adabas database.</p> <p>This clause is used to provide a password (<i>operand1</i>) when updating data from a file which is password-protected. The password (<i>operand1</i>) may be specified as an alphanumeric constant or as an alphanumeric variable. It may consist of up to 8 characters, and must not contain special characters or embedded blanks. If the password is specified as a constant, it must be enclosed in apostrophes.</p> <p>For further information, see the statements FIND and PASSW.</p>
CIPHER= <i>operand2</i>	<p>CIPHER Clause:</p> <p>The CIPHER clause is applicable only for an Adabas database.</p> <p>This clause is used to provide a cipher key (<i>operand2</i>) when updating data from a file which is enciphered. The cipher key (<i>operand2</i>) may be specified as an numeric constant with 8 digits or as a user-defined variable with format/length N8.</p> <p>For further information, see the statement FIND.</p>
USING NUMBER <i>operand3</i>	<p>USING NUMBER Clause:</p> <p>This clause can only be used for an Adabas database.</p>
GIVING NUMBER <i>operand3</i>	<p>GIVING NUMBER Clause:</p> <p>This clause is used to store a record with a user-supplied Adabas ISN. If a record with the specified ISN already exists, an error message will be returned and the execution of the program will be terminated unless ON ERROR processing was specified.</p>

Syntax Element	Description
SET/WITH <i>operand4=operand5</i>	<p>SET/WITH Clause:</p> <p>SET/WITH can be used in reporting mode to specify the fields for which values are being provided. Any field defined in the file that is not specified in the SET clause will contain a null value in the new record.</p> <p>This clause is not permitted if a DEFINE DATA statement is used, because in that case the STORE statement always refers to the entire view as defined in the DEFINE DATA statement.</p>
USING SAME (<i>r</i>)	<p>USING SAME Clause:</p> <p>In reporting mode, this clause can be used to indicate that the same field values as read in the statement referenced by the STORE statement (FIND, GET, READ) are to be used to add a new record.</p> <p>The statement reference notation (<i>r</i>) may be specified as a source-code line number or as a statement label.</p> <p>This clause is not permitted if a DEFINE DATA statement is used, because in that case the STORE statement would always refers to the entire view, as defined in the DEFINE DATA statement.</p>

Example

```

** Example 'STOEX1S': STORE (structured mode)
**
** CAUTION: Executing this example will modify the database records!
*****
DEFINE DATA LOCAL
1 EMPL-VIEW VIEW OF EMPLOYEES
  2 PERSONNEL-ID
  2 NAME
  2 FIRST-NAME
  2 MAR-STAT
  2 BIRTH
  2 CITY
  2 COUNTRY
*
1 #PERSONNEL-ID (A8)
1 #NAME (A20)
1 #FIRST-NAME (A15)
1 #BIRTH-D (D)
1 #MAR-STAT (A1)
1 #BIRTH (A8)
1 #CITY (A20)
1 #COUNTRY (A3)
1 #CONF (A1)
END-DEFINE
*
REPEAT
  INPUT 'ENTER A PERSONNEL ID AND NAME (OR ''END'' TO END)' //
    'PERSONNEL-ID : ' #PERSONNEL-ID //
    'NAME : ' #NAME /

```

```

        'FIRST-NAME      : ' #FIRST-NAME
/*
/*  VALIDATE ENTERED DATA
/*
IF #PERSONNEL-ID = 'END' OR #NAME = 'END'
    STOP
END-IF
IF #NAME = ' '
    REINPUT WITH TEXT 'ENTER A LAST-NAME' MARK 2 AND SOUND ALARM
END-IF
IF #FIRST-NAME = ' '
    REINPUT WITH TEXT 'ENTER A FIRST-NAME' MARK 3 AND SOUND ALARM
END-IF
/*
/*  ENSURE PERSON IS NOT ALREADY ON FILE
/*
FIND NUMBER EMPL-VIEW WITH PERSONNEL-ID = #PERSONNEL-ID
IF *NUMBER > 0
    REINPUT 'PERSON WITH SAME PERSONNEL-ID ALREADY EXISTS'
        MARK 1 AND SOUND ALARM
END-IF
MOVE 'N' TO #CONF
/*
/*  GET FURTHER INFORMATION
/*
INPUT
'ADDITIONAL PERSONNEL DATA'                ////
'PERSONNEL-ID          : ' #PERSONNEL-ID (AD=IO) /
'NAME                  : ' #NAME          (AD=IO) /
'FIRST-NAME            : ' #FIRST-NAME   (AD=IO) ///
'MARITAL STATUS        : ' #MAR-STAT     /
'DATE OF BIRTH (YYYYMMDD) : ' #BIRTH     /
'CITY                  : ' #CITY         /
'COUNTRY (3 CHARACTERS) : ' #COUNTRY     //
'ADD THIS RECORD (Y/N)  : ' #CONF        (AD=M)
/*
/*  ENSURE REQUIRED FIELDS CONTAIN VALID DATA
/*
IF NOT (#MAR-STAT = 'S' OR = 'M' OR = 'D' OR = 'W')
    REINPUT TEXT 'ENTER VALID MARITAL STATUS S=SINGLE ' -
        'M=MARRIED D=DIVORCED W=WIDOWED' MARK 1
END-IF
IF NOT (#BIRTH = MASK(YYYYMMDD) AND #BIRTH = MASK(1582-2699))
    REINPUT TEXT 'ENTER CORRECT DATE' MARK 2
END-IF
IF #CITY = ' '
    REINPUT TEXT 'ENTER A CITY NAME' MARK 3
END-IF
IF #COUNTRY = ' '
    REINPUT TEXT 'ENTER A COUNTRY CODE' MARK 4
END-IF
IF NOT (#CONF = 'N' OR = 'Y')
    REINPUT TEXT 'ENTER Y (YES) OR N (NO)' MARK 5
END-IF
IF #CONF = 'N'
    ESCAPE TOP
END-IF
/*
/*  ADD THE RECORD
/*
MOVE EDITED #BIRTH TO #BIRTH-D (EM=YYYYMMDD)
/*

```

```

EMPL-VIEW.PERSONNEL-ID := #PERSONNEL-ID
EMPL-VIEW.NAME         := #NAME
EMPL-VIEW.FIRST-NAME  := #FIRST-NAME
EMPL-VIEW.MAR-STAT    := #MAR-STAT
EMPL-VIEW.BIRTH       := #BIRTH-D
EMPL-VIEW.CITY        := #CITY
EMPL-VIEW.COUNTRY     := #COUNTRY
/*
STORE RECORD IN EMPL-VIEW
/*
END OF TRANSACTION
/*
WRITE NOTITLE 'RECORD HAS BEEN ADDED'
/*
END-REPEAT
END

```

Output of Program STOEX1S:

ENTER A PERSONNEL ID AND NAME (OR 'END' TO END)

PERSONNEL-ID : 90001100

NAME : JONES
FIRST-NAME : EDWARD

After entering and confirming the personnel key data, additional personnel data fields are displayed for input:

ADDITIONAL PERSONNEL DATA

PERSONNEL-ID : 90001100
NAME : JONES
FIRST-NAME : EDWARD

MARITAL STATUS :
DATE OF BIRTH (YYYYMMDD) :
CITY :
COUNTRY (3 CHARACTERS) :

ADD THIS RECORD (Y/N) : N

Equivalent reporting-mode example: STOEX1R.