# Read Statement

## Description

Reads a record from a native table, assigning the record to the specified variable.

## Syntax

**Read** *variable* From [*table_var* | Cursor *cursorvar*], *key* Then | Else *statements*

## Parameters

The Read statement has the following parameters.

| Parameter | Description |
|---|---|
| *Variable* | Assigned the value of the data that is read from the row specified by key. |
| *Table_var* | A table variable created by a previous Open statement. |
| *Cursorvar* | If accessing a cursor, cursorvar contains a cursor variable. Cursor variables are initialized with a Select...By statement and must be preceded with the word Cursor. If the table being accessed has had control features added, cursor access will automatically invoke domain conversion during a Read. |
| *Key* | The row referenced by key will be read from the table identified by table_var or the table accessed using the cursor in cursorvar. |
| *Then* | The statement(s) following Then are executed if a row key is read successfully. |
| *Else* | The statement(s) following Else are executed if the row in variable cannot be read. The Status() function indicates the severity of the error, and the system variable **@FILE_ERROR** contains details about the nature of the error. |

## See also

Open, ReadO, ReadV, MatRead, Write

## Example

```
/* The row having key 101 is read into the variable CUST as a dynamic array. */
open "CUSTOMER" To CUSTOMER_TABLE then
  read CUST From CUSTOMER_TABLE, "101" else
     status = Set_FSError()
     return
  end
end

/* Select...By initializes a Cursor with a sorted list of customer records. A ReadNext and Read loop reads each
record in turn,
passing the customer records to a local subroutine (not shown) for processing. Note the use of the Cursor
keyword and the Cursorvar. */

CURSOR_NO = ""
Select CUSTOMER_TABLE By "ST" Setting CURSOR_NO Else
  status = Set_FSError()
  Return
end
Done = 0
loop
  ReadNext @ID Using CURSOR_NO else Done = 1
Until Done Do
  read @RECORD From CUSTOMER_TABLE, @ID then
     status = Set_FSError()
     return
  end
  * processing logic here ...
  GoSub PROCESS
Repeat
```