

# ReadO Statement

## Description

Use to read a record that you do not intend to modify.

## Syntax

**ReadO** *variable* *From* [*table\_var* | *Cursor cursorvar*], *key* *Then* | *Else* statements

## Parameters

The ReadO statement has the following parameters.

Parameter	Description
<i>variable</i>	Assigned the value of the data from the specified column of the row specified by key.
<i>Table_var</i>	A table variable that has been previously specified in an <a href="#">Open statement</a> .
<i>cursorvar</i>	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 ReadO.
<i>Key</i>	The row referenced by key will be read from the table identified by table_var or the table accessed using the cursor in cursorvar.
<i>Then</i>	The statements in the Then clause will be executed if a row is read successfully.
<i>Else</i>	The statement(s) following Else are executed if the row in variable cannot be read. The <a href="#">Status()</a> function indicates the severity of the error, and the system variable <b>@FILE_ERROR</b> contains details about the nature of the error.

## Note

ReadO is used to read a row that will not be updated. As a consequence, the read request may be fulfilled from any cache or buffers being maintained by the filing system. ReadO, therefore, can be a faster method of accessing data than by using the [Read statement](#). In a multi-user environment, ReadO does not necessarily read the most current version of a row.

**Note:** The definition of ReadO (and the use of caches or buffers) is specific to the filing system being used. Many filing systems do not cache data and therefore implement [Read](#) and ReadO identically.

## See also

[MatRead](#), [Open](#), [Read](#), [Write](#), [Xlate\(\)](#)

## Example

```
/* This code fragment counts the number of bytes in the CUSTOMERS table. The result is stored in data_bytes at
the end of the loop.
ReadO is used because record locking is unnecessary, since the only purpose of reading each row is to determine
its length. */

Open "CUSTOMERS" To customers_file Then
data_bytes = 0
Select customers_file
Done = 0
Loop
  ReadNext @id Else Done = 1
Until Done do
  ReadO @record From customers_file, @id Then
    data_bytes += len(@record) + len(@id)
  End
Repeat
/* data_bytes contains the number of data bytes in the table. */
End
```

