

OSBRead Statement

Description

Reads from an operating system file opened with OSOpen.

Syntax

OSBRead *variable* From *filevar* At *byte* Length *length*

Parameters

The OSBRead statement has the following parameters.

Parameter	Description
<i>variable</i>	The variable into which the file portion will be read.
<i>filevar</i>	The variable to which the operating system file was assigned when it was opened with an OSOpen statement.
<i>byte</i>	An integer identifying how many bytes (offset) into the operating system file the read operation should begin. A 0 (zero) integer indicates the read should start at the beginning of the file.
<i>length</i>	An integer that tells how many bytes to read.

After the execution of an OSBRead statement, the Status() of the read is returned with one of the following codes:

Value	Meaning
0	No error.
1	Bad OS filename.
2	Access denied by operating system.
3	Disk or directory full.
4	File does not exist.
5	Unknown error.
6	Attempt to write to a read-only file.

Note: *There is no Then...Else statement for OSBRead.*

See also

[OSBWrite](#), [OSClose](#), [OSDelete](#), [OSOpen](#), [OSRead](#), [OSWrite](#)

Example

```

/* This code reads an existing OS file and copies it in 100 character chunks to a new OS file */

Equ RECSIZE$ To 100
readOffset = 0
writeOffset = 0

filename = "c:\temp\my_data.txt"
newFileName = "c:\temp\my_new_data.txt"

oswrite "" To newFileName ; * create the new file

OSOpen filename To inputFileHandle then
  OSOpen newFileName To outputFileHandle Then
    Loop
      OSBRead data From inputFileHandle At readOffset length RECSIZE$
      error = status()
    Until data = NULL$
      readOffset += RECSIZE$
      OSBWrite data On outputFileHandle At writeOffset
      writeOffset += RECSIZE$
    Repeat
  end else
    error = status()
  end
End else
  error = status()
End

osclose inputFileHandle
osclose outputFileHandle

```

Example 2

```

/* The following example shows how to use BASIC+ to handle operating system files. */
* Open the test file, or, if it does not exist, create it.
OSOpen "OSTEST" To file_var Else
  * Create an operating system file.
  OSWrite "" To "OSTEST"
  * Create the necessary file handle.
  OSOpen "OSTEST" To file_var Else
    * error handling
  End
End
/* You might have reason to check a particular segment of data in an operating system file, as for example when
looking at a data file header for which you know the length. In the following line, the desired segment is
presumed to be 256 bytes in length, beginning at the first byte of the file. */
OSBRead data_segment From file_var At 0 length 256
/* Having the 256-byte segment within OpenInsight for Workgroups, you can now check particular locations for
required information. The following code looks for the string "Revelation Software" within the 256-byte
segment. */
search = "Revelation Software"
start = Index(data_segment, search, 1)
if start then
  found = data.segment [start, len(search)]
  * The found string is converted to uppercase
  Convert @LOWER_CASE To @UPPER_CASE in found_data
  /* The converted string is now written directly back to the OS file. */
  OSBWrite found_data On file_var at start
end
OSClose file_var

```