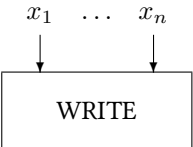


Block WRITE

The WRITE block writes input data to a file with sequential access.



Name	WRITE
Function	fb0017
Inputs	1 ... [200]
Outputs	0
Parameters	0 ... [3]
Strings	0 ... [3]
Group	S

Inputs

- 1 Any value x_1 to be written to file fn (Inputs for file name qualifiers are ignored)
- 2 Any value x_2 to be written to file fn
- n Any value x_n to be written to file fn with $n \leq 200$

Outputs

None

Parameters

- 1 Mode
 - 0 If file fn already exists, an error message is given (default)
 - 1 If file fn already exists, it is overwritten
 - 2 If file fn already exists, new records are appended
- 2 Suppress file name qualifier inputs (Deprecated! This parameter was basically never used, and can be safely ignored)
 - 0 No (default)
 - 1 Yes
- 3 Separator (Deprecated!)
 - 0 Blank (default)
 - 1 Comma

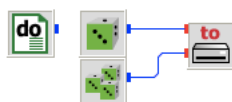
2 Semicolon

Strings

- 1 File name fn. If the first character of the file name without path) is a # the name is parsed for variable qualifiers – see Description for further details. If SP(1) is empty a default file `insel.tmp` is generated with in the temporary INSEL directory.
- 2 Fortran format
- 3 Optional headline

Remarks JS 8 Dec 2016: Parameters for non-space separators and optional headline added. However, works only for `.insel` files (`.vseit` does not support empty string parameters) and implemented only for filenames without qualifiers.

Remarks Integers must be written in format `Fx.0`, where x is the number of bytes.
`write.vseit`



A **DO** block counts from 1 to 8760 (the number of hours of a non-leap year) and two different random number sets are generated by the blocks **GASDEV** and **RAN1**. The **WRITE** block writes these numbers to a data file named `myRandom.dat`.

The first records of the resulting file are these:

```
9.92046744E-02  1.6216065
1.86635051E-02 -0.39489648
0.80818379     -0.33821103
0.75034720     0.53852010
0.79094255     -0.42136794
```

Of course, it is necessary to have write access to the directory where the output file shall be located.

Variable file names If the first character of a specified parameter is a # character the file name is considered variable and is parsed by INSEL. Up to six qualifiers can be used as place holders for digits. The qualifiers have the format `%nx`, where n is a positive digit in the range of 1 to 9. x is a character out of `YMDhms` – reminding on their most frequent use of date and time information in data file names.

The numerical values for the n bytes must be provides as block inputs. Y is used with the first input, M with the second, and so forth.

Example The file name `#myName%4Y.dat` results in `myName2011.dat`, assuming that the first input has a value of 2011.

Adding a second input which contains values for the months of a simulation, a qualified file name would be #myName%4Yplus%2M.dat. For a simulation running over a complete year the WRITE block creates files with names myName2011plus1.dat, myName2011plus2.dat, ... myName2011plus11.dat, myName2011plus12.dat.

If the months from January to September shall be written with leading zeros the qualifier #myName%4Yplus%2.2M.dat is accepted and creates files myName2011plus01.dat, myName2011plus02.dat, and so forth. The same rules apply to all blocks which read data, too.

See also Blocks [READ](#), [READD](#), and [READN](#).