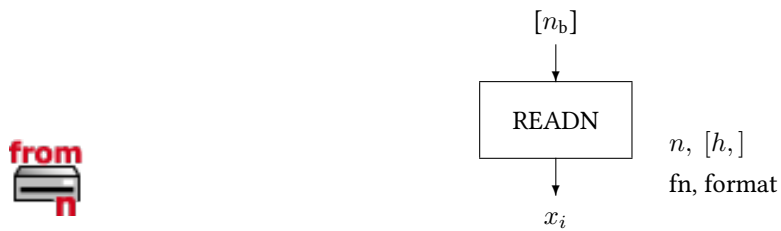


Block READN

The READN reads n data from a file with sequential access and successively provides them as outputs; after n calls READN performs a new read.



Name	READN
Function	fb0038
Inputs	0 . . . [5]
Outputs	1
Parameters	2
Strings	2
Group	S

Inputs

- 1 Output of a predecessor (optional). If from one time step to the next the value decreases the block performs a Rewind of the file.
- 2 File name qualifier
- 3 File name qualifier
- 4 File name qualifier
- 5 File name qualifier

Outputs

- 1 Any value x_i read from file fn

Parameters

- 1 Number n of values to be read per record
- 2 Number h of records to be skipped on the first call (file header, the default is $h = 0$)

Strings

- 1 File name fn
- 2 Fortran format

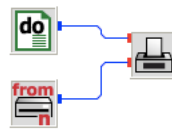
Remarks Integers must be read in format Fx.0, where x is the number of bytes.

All records of file fn must have the same physical record length.

dwd2002.dat The next lines show (a part of) the first records of a data file `dwd2002.dat` which can be found in the `examples/data` directory. The file contains irradiance data in kWh per m² and month, as published by the German Weather Service DWD for the year 2002.

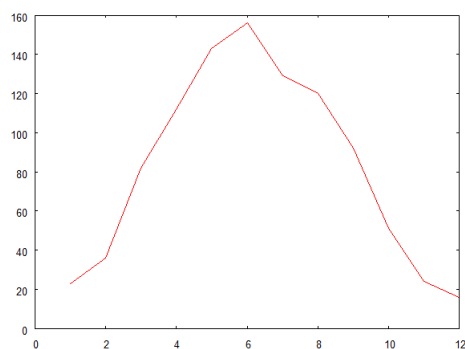
Aachen	23	35	88	123	142	...	16	1001	-1
Augsburg	34	47	100	127	152	...	22	1178	3
Bad Lippspringe	22	34	79	111	135	...	16	928	-3
Berlin	21	38	68	103	144	...	17	1004	-1
Bocholt	20	40	90	119	139	...	14	1033	3
Bochum	23	36	82	112	143	...	16	981	2
Bonn	26	41	87	121	142	...	16	1005	1
Braunlage	20	32	65	104	133	...	13	938	-2

readn.vseit



A **READN** block is used to read data from the file `dwd2002.dat`. The **Number of records to be skipped on the first call** is set to five, so that the block actually reads the data for the location Bochum. The number of values to be read per record is set to twelve, with sequential access. In this case the Fortran format (20X, 12F6.0, 12X) is used to read the data.

The 'clicks' are generated by a **DO** block. A **PLOT** block displays the twelve monthly means.



See also Blocks **READ**, **READD**, and **WRITE**.