

	example
	example
refmt)	example

A as 8-bit unsigned integers to a binary file in column-major order. Use `fopen` to open the file and obtain the file ID. Use `fwrite` to write the data to the file and `fclose` to close the file by calling `fclose(fileID)`.

example

as in A in the form and size described by precision.

example

number of bytes or bits specified by skip before writing

`format`) additionally specifies the order for writing bytes or

example

elements of A that `fwrite` successfully writes to the file. Arguments of the previous syntaxes.

collapse all

to write access using 'w' in the call to `fopen`.

double integers.

are. Use the precision argument, 'integer*4', to write 4-byte

;

the 4-by-4 magic square, stored as double-precision floating-point

that enables appending to the file. Specify the file-access type, 'a', in

en, close the file.

ng

file named myfile.bin for use on a big-endian system. Specify a
fwrite, to indicate big-endian byte ordering.

3-be');

[collapse all](#)

d from fopen, 1 for standard output (the screen), or 2 for standard

cter array.

int32 | int64 | uint8 | uint16 | uint32 | uint64 | logical | char

write

pecified as one of the following strings in the Precision column.

sion	Bits (Bytes)
	32 (4)
	8 (1)
6	16 (2)
2	32 (4)
4	64 (8)
	8 (1)
ned char	8 (1)
t	16 (2)
	32 (4)
	1 ≤ n ≤ 64
	32 (4)
	8 (1)
	16 (2)
	32 (4)
	64 (8)
er*1	8 (1)
er*2	16 (2)
er*4	32 (4)
er*8	64 (8)
	8 (1)
d char	8 (1)
	16 (2)
	32 (4)

	$1 \leq n \leq 64$
e	32 (4)
e	64 (8)
	32 (4)
32	32 (4)
64	64 (8)
4	32 (4)
8	64 (8)
1	8 (1)
	Depends on the encoding scheme associated with the file. Set encoding with fopen.

an fwrite saturates for all values outside the range.

ATLAB®, read and write data of class double or single.

alue, specified as a scalar. If you specify a precision of bitn or

contiguous fields in fixed-length records.

l as one of the strings in the table that follows. For bitn and ubitn
r writing bits within a byte, but the order for writing bytes remains your

Your system byte ordering (default)
Big-endian ordering
Little-endian ordering
Big-endian ordering, 64-bit long data type
Little-endian ordering, 64-bit long data type

se little-endian ordering for new files. Existing binary files can use

;

n

anf | fseek | ftell
