CONTENTS Close

example

example

) example

,machinefmt) example

example

inary file into column vector A and positions the file
is indicated by the file identifier, fileID. Use fopen to
n you finish reading, close the file by calling

an array, A, with dimensions, sizeA, and positions the
lates A in column order.

ets values in the file according to the form and size example
s optional.

skips the number of bytes or bits specified by skip after example
ent is optional.

achinefmt) additionally specifies the order for reading example
uments are optional.

the number of characters that fread reads into A. You
nts of the previous syntaxes.

collapse all

ine.bin.

ss double. By default, fread reads a file 1 byte at a time, interprets
), and returns a double array.

ent for each byte in the file.

```
```

```
:lass     Attributes

louble
```

```
```

ıing nine double-precision values.

```
);
```

data in the file into a 3-by-3 array, A. Specify that the source data is

```
```

```
```

ose the output array, A so that it is a row vector.

```
```

values from 1 to 9. Write the data as uint16 values.

```
```

pecify that the source data is class uint16.

```
```

e with the first six values from the file, nine.bin.

```
```

e before reading the next values. Specify this format using the
data is class uint16, one value is represented by 2 bytes. Therefore,

```
ip)
```

ın-wise with the values from `nine.bin`.

) values.

;

ınd. If you read 4 bits at a time on a little-endian system, your results

ind. Read the data 4 bits at a time as before, but specify a big-endian

```
ieee-be');
```

collapse all

as an integer. Before reading a file with `fread`, you must use `fopen` to

or

s `Inf`, an integer, or a two-element row vector.

	Dimensions of the output array, `A`
	Column vector, with each element containing a value in the file.
	Column vector with `n` elements.
	`m`-by-`n` matrix, filled in column order. `n` can be `Inf`, but `m` cannot.

ead

ecified as a string in one of the following forms. Optionally the input

	Description
	Input values are of the class specified by *source*. Output matrix A is class double. Example: 'int16'
	Input values are of the class specified by *source*. The class of the output matrix, A, is specified by *output*. Example: 'int8=>char'
	The input values and the output matrix, A, are of the class specified by *source*. For bit*n* or ubit*n* precisions, the output has the smallest class that can contain the input. Example: '*ubit18' This is equivalent to 'ubit18=>uint32'
	Read *N* values before skipping the number of bytes specified by the skip argument. Example: '4*int8'

source and *output*.

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 ≤ n ≤ 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`.

ne end of the file before reading a complete value, it does not return a
s `bit`*n* or `ubit`*n*, then `fread` returns a partial result for the final value.

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

lue, specified as a scalar. If you specify a `precision` of `bit`*n* or `ubit`*n*,

contiguous fields in fixed-length records.

s one of the strings in the table that follows. For `bit`*n* and `ubit`*n*
r reading bits within a byte, but the order for reading bytes remains

	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

collapse all

specified the `sizeA` argument, then `A` is a matrix of the specified size.
different class in the `precision` argument.

lar value.

---

ek | `ftell` | `fwrite`
