



MatrixMarket I/O Functions for Matlab®

Matrix Market provides Matlab® M-files for three basic Matrix Market file I/O functions: **mminfo**, **mmread**, and **mmwrite**. Their use is described here.

File: [mminfo.m](#)

Requires: `strtok.m` from Matlab Version 4.2.

If using a previous version of Matlab, substitute [gettok.m](#)

Function: `[rows, cols, entries, rep, field, symm] = mminfo(filename);`

Description: Queries the contents of the Matrix Market file 'filename' to extract size and storage information.

In the case of coordinate matrices, `entries` refers to the number of coordinate entries stored in the file, since the number of non-zero entries in the final matrix cannot be determined until the data is read (and symmetrized, if necessary).

In the case of array matrices, `entries` is the product `rows*cols`, regardless of whether symmetry was used to store the matrix efficiently.

File: [mmread.m](#)

Requires: `strtok.m` from Matlab Version 4.2.

If using a previous version of Matlab, substitute [gettok.m](#)

Function: `[A, rows, cols, entries] = mmread(filename);`

Description: Reads the contents of the Matrix Market file 'filename' into the matrix 'A'. 'A' will be either sparse or full (in the Matlab sense) depending on the Matrix Market format, indicated by 'coordinate' (coordinate sparse storage) or 'array' (dense array storage). The data will be duplicated as appropriate if symmetry is indicated in the header.

Optionally, size information about the matrix can be obtained by using the return values `rows`, `cols`, and `entries`, where `entries` is the number of nonzero entries in the final matrix.

File: [mmwrite.m](#)

Function: `[err] = mmwrite(filename,A,comment,field,precision);`

Description: Writes the sparse or dense matrix A to a Matrix Market formatted file.

Arguments:

(required) `filename` Matrix Market filename (e.g. 'young1c.mtx')

(required) `A` sparse or full matrix

(optional) `comment` Matrix of comments to prepend to the Matrix Market file. To build a comment matrix, use `str2mat`. For example:

```
comment = str2mat(' Comment 1' , ...
' Comment 2' , ...
' and so on.' , ...
' to attach a date:' , ...
[' ',date]);
```

The default comment is a single line date stamp.

(optional) `field` 'real'
'complex'
'integer'
'pattern'
If omitted, data will determine type.

(optional) `precision` Number of digits to display for real or complex values.
If omitted, full working precision is used.