

[Contents](#)   [Description](#)   [Remarks and examples](#)   [Reference](#)   [Also see](#)

## Contents

[M-5] Manual entry	Function	Purpose
<b>Console output</b>		
<b>printf()</b>	<code>printf()</code> <code>sprintf()</code>	display display into string
<b>errprintf()</b>	<code>errprintf()</code>	display error message
<b>display()</b>	<code>display()</code>	display text interpreting SMCL
<b>displayas()</b>	<code>displayas()</code>	set whether output is displayed
<b>displayflush()</b>	<code>displayflush()</code>	flush terminal output buffer
<b>liststruct()</b>	<code>liststruct()</code>	list structure's contents
<b>more()</b>	<code>more()</code> <code>setmore()</code> <code>setmoreonexit()</code>	create <code>—more—</code> condition query or set more on or off set more on or off on exit
<b>File directories</b>		
<b>direxists()</b>	<code>direxists()</code>	whether directory exists
<b>dir()</b>	<code>dir()</code>	file list
<b>chdir()</b>	<code>pwd()</code> <code>chdir()</code> <code>mkdir()</code> <code>rmdir()</code>	obtain current working directory change current working directory make new directory remove directory
<b>File management</b>		
<b>findfile()</b>	<code>findfile()</code>	find file
<b>fileexists()</b>	<code>fileexists()</code>	whether file exists
<b>cat()</b>	<code>cat()</code>	read file into string matrix
<b>unlink()</b>	<code>unlink()</code>	erase file
<b>adosubdir()</b>	<code>adosubdir()</code>	obtain ado-subdirectory for file
<b>issamefile()</b>	<code>issamefile()</code>	whether two file paths point to the same file

<b>fopen()</b>	fopen()	open file
	fclose()	close file
	fget()	read line of text file
	fgetnl()	same, but include newline character
	fread()	read <i>k</i> bytes of binary file
	fput()	write line into text file
	fwrite()	write <i>k</i> bytes into binary file
	fgetmatrix()	read matrix
	fputmatrix()	write matrix
	fstatus()	status of last I/O command
	ftell()	report location in file
	fseek()	seek to location in file
	ftruncate()	truncate file at current position
<b>ferrortext()</b>	ferrortext()	error text of file error code
	freturncode()	return code of file error code
<b>bufio()</b>	bufio()	initialize buffer
	bufbyteorder()	reset (specify) byte order
	bufmissingvalue()	reset (specify) missing-value encoding
	bufput()	copy into buffer
	bufget()	copy from buffer
	fbufput()	copy into and write buffer
	fbufget()	read and copy from buffer
	bufbfmtlen()	utility routine
	bufbfmtisnum()	utility routine
<b>xl()</b>	xl()	Excel file I/O class
<b>_docx*()</b>	_docx*()	generate Office Open XML file
<b>Pdf*()</b>	Pdf*()	create a PDF file

Filename & path manipulation
------------------------------

<code>pathjoin()</code>	<code>pathjoin()</code>	join paths
	<code>pathsplit()</code>	split paths
	<code>pathbasename()</code>	path basename
	<code>pathsuffix()</code>	file suffix
	<code>pathrmsuffix()</code>	remove file suffix
	<code>pathisurl()</code>	whether path is URL
	<code>pathisabs()</code>	whether path is absolute
	<code>pathasciisuffix()</code>	whether file is text
	<code>pathstatastatusuffix()</code>	whether file is Stata
	<code>pathlist()</code>	process path list
	<code>pathsubsysdir()</code>	substitute for system directories
	<code>pathsearchlist()</code>	path list to search for file
	<code>pathresolve()</code>	resolve a relative path
	<code>pathgetparent()</code>	get the parent path

## Description

The above functions have to do with

1. Displaying output at the terminal.
2. Reading and writing data in a file.

## Remarks and examples

[stata.com](http://stata.com)

To display the contents of a scalar, vector, or matrix, it is sufficient merely to code the identity of the scalar, vector, or matrix:

```
: x
      1      2      3      4
1   .1369840784   .643220668   .5578016951   .6047949435
```

You can follow this approach even in programs:

```
function example()
{
    ...
    "i am about to calculate the result"
    ...
    "the result is"
    b
}
```

On the other hand, `display()` and `printf()` (see [M-5] [display\(\)](#) and [M-5] [printf\(\)](#)) will allow you to exercise more control over how the output looks.

Changing the subject: you will find that many I/O functions come in two varieties: with and without an underscore in front of the name, such as `_fopen()` and `fopen()`. As always, functions that begin with an underscore are generally silent about their work and return flags indicating their success or failure. Functions that omit the underscore abort and issue the appropriate error message when things go wrong.

## Reference

Gould, W. W. 2009. [Mata Matters: File processing](#). *Stata Journal* 9: 599–620.

## Also see

[M-4] [Intro](#) — Categorical guide to Mata functions

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2023 StataCorp LLC, College Station, TX, USA. All rights reserved.

