

**unlink()** — Erase file

|                            |                    |                      |                |
|----------------------------|--------------------|----------------------|----------------|
| Description<br>Diagnostics | Syntax<br>Also see | Remarks and examples | Conformability |
|----------------------------|--------------------|----------------------|----------------|

## Description

`unlink(filename)` erases *filename* if it exists, does nothing if *filename* does not exist, and aborts with error if *filename* exists but cannot be erased.

`_unlink(filename)` does the same, except that, if *filename* cannot be erased, rather than aborting with error, `_unlink()` returns a negative error code. `unlink()` returns 0 if *filename* was erased or *filename* did not exist.

## Syntax

*void*                    `unlink(string scalar filename)`

*real scalar*   `_unlink(string scalar filename)`

## Remarks and examples

[stata.com](#)

To remove directories, see `rmdir()` in [M-5] [chdir\(\)](#).

## Conformability

`unlink(filename)`  
*filename:*     $1 \times 1$   
*result:*        *void*

`_unlink(filename)`  
*filename:*     $1 \times 1$   
*result:*         $1 \times 1$

## Diagnostics

`unlink(filename)` aborts with error when `_unlink()` would give a negative result.

`_unlink(filename)` returns a negative result if the file cannot be erased and returns 0 otherwise. If the file did not exist, 0 is returned. When there is an error, most commonly returned are -3602 (filename invalid) or -3621 (file is read-only).

## Also see

[M-4] [IO](#) — I/O functions