



**NAME**

splint – A tool for statically checking C programs

**SYNOPSIS**

**splint** [options]

**DESCRIPTION**

**Splint** is a tool for statically checking C programs for security vulnerabilities and common programming mistakes. With minimal effort, Splint can be used as a better lint(1). If additional effort is invested adding annotations to programs, Splint can perform stronger checks than can be done by any standard lint. For full documentation, please see <http://www.splint.org>. This man page only covers a few of the available options.

**OPTIONS**

**-help** Shows help

**Initialization**

These flags control directories and files used by Splint. They may be used from the command line or in an options file, but may not be used as control comments in the source code. Except where noted, they have the same meaning preceded by `-` or `+`.

**-tmpdir** *directory*  
Set directory for writing temp files. Default is `/tmp/`.

**-I** *directory*  
Add directory to path searched for C include files. Note there is no space after the I, to be consistent with C preprocessor flags.

**-S** *directory*  
Add directory to path search for `.lcl` specification files.

**-f** *file* Load options file `<file>`. If this flag is used from the command line, the default `~/splintrc` file is not loaded. This flag may be used in an options file to load in another options file.

**-nof** Prevents the default options files (`./splintrc` and `~/splintrc`) from being loaded. (Setting `-nof` overrides `+nof`, causing the options files to be loaded normally.)

**-systemdirs** *directories*  
Set directories for system files (default is `/usr/include`). Separate directories with colons (e.g., `/usr/include:/usr/local/lib`). Flag settings propagate to files in a system directory. If `-systemdi-errors` is set, no errors are reported for files in system directories.

**Pre-processor**

These flags are used to define or undefine pre-processor constants. The `-I<directory>` flag is also passed to the C pre-processor.

**-D** *initializer*  
Passed to the C pre-processor.

**-U *initializer***

Passed to the C pre-processor

**Libraries** These flags control the creation and use of libraries.

**-dump *file***

Save state in <file> for loading. The default extension .lcd is added if <file> has no extension.

**-load *file***

Load state from <file> (created by -dump). The default extension .lcd is added if <file> has no extension. Only one library file may be loaded.

By default, the standard library is loaded if the -load flag is not used to load a user library. If no user library is loaded, one of the following flags may be used to select a different standard library. Precede the flag by + to load the described library (or prevent a library from being loaded using nolib). See Appendix F for information on the provided libraries.

**-nolib**

Do not load any library. This prevents the standard library from being loaded.

**-ansi-lib**

Use the ANSI standard library (selected by default).

**-strict-lib**

Use strict version of the ANSI standard library.

**-posix-lib**

Use the POSIX standard library.

**-posix-strict-lib**

Use the strict version of the POSIX standard library.

**-1-lib** Use UNIX version of standard library.

**-1-strict-lib**

Use the strict version of the UNIX standard library.

**Output**

These flags control what additional information is printed by Splint. Setting +<flag> causes the described information to be printed; setting -<flag> prevents it. By default, all these flags are off.

**-usestderr**

Send error messages to standard error (instead of standard out).

**-showsummary**

Show a summary of all errors reported and suppressed. Counts of suppressed errors are not necessarily correct since turning a flag off may prevent some checking from being done to save

computation, and errors that are not reported may propagate differently from when they are reported.

**-showscan**

Show file names as they are processed.

**-showalluses**

Show list of uses of all external identifiers sorted by number of uses.

**-stats** Display number of lines processed and checking time.

**-timedist**

Display distribution of where checking time is spent.

**-quiet**

Suppress herald and error count. (If quiet is not set, Splint prints out a herald with version information before checking begins, and a line summarizing the total number of errors reported.)

**-whichlib**

Print out the standard library filename and creation information.

**-limit *number***

At most <number> similar errors are reported consecutively. Further errors are suppressed, and a message showing the number of suppressed messages is printed.

**Expected Errors**

Normally, Splint will expect to report no errors. The exit status will be success (0) if no errors are reported, and failure if any errors are reported. Flags can be used to set the expected number of reported errors. Because of the provided error suppression mechanisms, these options should probably not be used for final checking real programs but may be useful in developing programs using make.

**-expect <number>**

Exactly <number> code errors are expected. Splint will exit with failure exit status unless <number> code errors are detected.

**-Message Format**

These flags control how messages are printed. They may be set at the command line, in options files, or locally in syntactic comments. The `linelen` and `limit` flags may be preceded by + or - with the same meaning; for the other flags, + turns on the describe printing and - turns it off. The box to the left of each flag gives its default value.

**-showcolumn**

Show column number where error is found. Default: +

**-showfunc**

Show name of function (or macro) definition containing error. The function name is printed once before the first message detected in that function. Default: +

**-showallconjs**

Show all possible alternate types (see Section 8.2.2). Default: -

**-paren-file-format**

Use file(line) format in messages.

**-hints** Provide hints describing an error and how a message may be suppressed for the first error reported in each error class. Default: +

**-forcehints**

Provide hints for all errors reported, even if the hint has already been displayed for the same error class. Default: -

**-linelen** *number*

Set length of maximum message line to <number> characters. Splint will split messages longer than <number> characters long into multiple lines. Default: 80

**Mode Selector Flags**

Mode selects flags set the mode checking flags to predefined values. They provide a quick coarse-grain way of controlling what classes of errors are reported. Specific checking flags may be set after a mode flag to override the mode settings. Mode flags may be used locally, however the mode settings will override specific command line flag settings. A warning is produced if a mode flag is used after a mode checking flag has been set.

These are brief descriptions to give a general idea of what each mode does. To see the complete flag settings in each mode, use splint -help modes. A mode flag has the same effect when used with either + or -.

**-weak**

Weak checking, intended for typical unannotated C code. No modifies checking, macro checking, rep exposure, or clean interface checking is done. Return values of type int may be ignored. The types bool, int, char and user-defined enum types are all equivalent. Old style declarations are unreported.

**-standard**

The default mode. All checking done by weak, plus modifies checking, global alias checking, use all parameters, using released storage, ignored return values or any type, macro checking, unreachable code, infinite loops, and fall-through cases. The types bool, int and char are distinct. Old style declarations are reported.

**-checks**

Moderately strict checking. All checking done by standard, plus must modification checking, rep exposure, return alias, memory management and complete interfaces.

**-strict**

Absurdly strict checking. All checking done by checks, plus modifications and global variables used in unspecified functions, strict standard library, and strict typing of C operators. A special reward will be presented to the first person to produce a real program that produces no errors with strict checking.

**AUTHOR**

If you need to get in contact with the authors send email to

or visit

**SEE ALSO**

lint(1)