

CLISP manual page

- [Name](#)
 - [Synopsis](#)
 - [Description](#)
 - [Options](#)
 - [Reference](#)
 - [Use](#)
 - [Files](#)
 - [Environment](#)
 - [See also](#)
 - [Bugs](#)
 - [Projects](#)
 - [Authors](#)
-

Name

clisp - Common Lisp language interpreter and compiler

Synopsis

```
clisp [ -h | --help ] [ --version ] [ --license ] [ -B lisplibdir ] [ -K linking-set ] [ -M memfile ] [ -m memsize ] [ -L language ] [ -N localedir ] [ -Edomain encoding ] [ -q | --quiet | --silent ] [ -w ] [ -I ] [ -a ] [ -p packagename ] [ -C ] [ -norc ] [ -i initfile ... ] [ -c [ -l ] lispfile [ -o outputfile ]... ] [ -x expression ] [ lispfile [ argument ... ] ]
```

Description

Invokes the common lisp interpreter and compiler. Invoked without arguments, executes a read-eval-print loop, in which expressions are in turn read from standard input, evaluated by the lisp interpreter, and their results output to standard output. Invoked with [-c](#), the specified lisp files are compiled to a bytecode that can be executed more efficiently.

Options

-h, --help

Displays a help message on how to use clisp.

--version

Displays the clisp version number, as given by the function call ([lisp-implementation-version](#)).

--license

Displays a summary of the licensing information, the GNU GPL.

-B *lisplibdir*

Specifies the installation directory. This is the directory containing the linking sets and other data files. This option is normally not necessary, because the installation directory is already built-in in the `clisp` executable.

-K *linking-set*

Specifies the linking set to be run. This is a directory containing at least a main executable and an initial memory image. Possible values are `base`, `full`. The default is `base`.

-M *memfile*

Specifies the initial memory image. This must be a memory dump produced by the [saveinitmem](#) function. It may have been compressed using GNU `gzip`.

-m *memsize*

Sets the amount of memory `clisp` tries to grab on startup. The amount may be given as `nnnnnnnn` (measured in bytes), `nnnn K` or `nnnn KB` (measured in kilobytes) or `n M` or `n MB` (measured in megabytes). Default is 2 megabytes. The argument is constrained above 100 KB. -- This version of `clisp` is not likely to actually use the entire *memsize* since garbage collection will periodically reduce the amount of used memory. It is therefore common to specify 10 MB even if only 2 MB are going to be used.

-L *language*

Specifies the language `clisp` uses to communicate with the user. This may be `english`, `deutsch`, `francais`, `espanol`. Other languages may be specified through the environment variable `LANG`, provided the corresponding message catalog is installed.

-N *localedir*

Specifies the base directory of locale files. `clisp` will search its message catalogs in `localedir/language/LC_MESSAGES/clisp.mo`.

-E *domain encoding*

Specifies the encoding used for a given domain, overriding the default which depends on the environment variables `LC_ALL`, `LC_CTYPE`, `LANG`. *domain* can be `file`, affecting `*default-file-encoding*`, or `pathname`, affecting `*pathname-encoding*`, or `terminal`, affecting `*terminal-encoding*`, or `foreign`, affecting `*foreign-encoding*`, or `misc`, affecting `*misc-encoding*`.

-q, --quiet, --silent

Quiet: `clisp` displays no banner at startup and no good-bye message when quitting.

-w

Wait for keypress after program termination.

-I

Be ILISP friendly. ILISP is an alternative Emacs interface to Common Lisp systems. With this option, `clisp` interacts in a way that ILISP can deal with. Currently the only effect of this is that unnecessary prompts are not suppressed. Furthermore, the GNU readline library treats Tab as a normal self-inserting character.

-a

ANSI CL compliant: Comply with the ANSI CL specification even on those issues where ANSI

CL is broken. This option is provided for maximum portability of Lisp programs, and is not useful for actual everyday work. It sets the variable `*package*` to [COMMON-LISP-USER](#) and the symbol macro `*ansi*` to `t`. See [impnotes.html, section "Maximum ANSI CL compliance"](#), for details.

-p *packagename*

At startup the value of the variable `*package*` will be set to the package named *packagename*. The default is the package which was active when the image was [saved](#), normally [USER](#), or [COMMON-LISP-USER](#) if the option [-a](#) was specified.

-C

Compile when load: At startup the value of the variable `*load-compiling*` will be set to `t`. Code being loaded will then be compiled on the fly. This results in slower loading, but faster execution.

-norc

Normally `clisp` loads a user run control (RC) file on startup (this happens after the [-C](#) option is processed). The file loaded is `${HOME}/.clisprc.lisp` or `${HOME}/.clisprc.fas`, whichever is newest. This option, `-norc`, prevents loading of the RC file.

-i *initfile ...*

Specifies initialization files to be loaded at startup. These should be lisp files (source or compiled). Several `-i` options can be given; all the specified files will be loaded in order.

-c *lispfile ...*

Compiles the specified *lispfiles* to bytecode (`*.fas`). The compiled files can then be loaded instead of the sources to gain efficiency.

-o *outputfile*

Specifies the output file or directory for the compilation of the last specified *lispfile*.

-l

A bytecode listing (`*.lis`) of the files being compiled will be produced. Useful only for debugging purposes. See the documentation of [compile-file](#) for details.

-x *expressions*

Executes a series of arbitrary expressions instead of a read-eval-print loop. The values of the expressions will be output to standard output. Due to the argument processing done by the shell, the *expressions* must be enclosed in double quotes, and double quotes and backslashes must be preceded by backslashes.

***lispfile* [*argument ...*]**

Loads and executes a *lispfile*. There will be no read-eval-print loop. Before *lispfile* is loaded, the variable `*args*` will be bound to a list of strings, representing the *arguments*. The first line of *lispfile* may start with `#!`, thus permitting `clisp` to be used as a script interpreter. If *lispfile* is `-`, the standard input is used instead of a file. This option must be the last one. No [RC file](#) will be executed.

Reference

The language implemented mostly conforms to

ANSI Common Lisp standard X3.226-1994 http://www.x3.org/tc_home/j13sd4.htm

available online as the

Common Lisp HyperSpec <http://www.harlequin.com/education/books/HyperSpec/>
("CLHS" for short)

which supersedes the earlier specifications

Guy L. Steele Jr.: Common Lisp - The Language. Digital Press. 2nd edition 1990, 1032 pages. <http://www.cs.cmu.edu/afs/cs.cmu.edu/project/ai-repository/ai/html/cltl/cltl2.html>
("CLtL2" for short)

and

Guy L. Steele Jr.: Common Lisp - The Language. Digital Press. 1st edition 1984, 465 pages. ("CLtL1" for short)

Use

help

to get some on-line help.

(*apropos name*)

lists the symbols relating to *name*.

(*exit*) or (*quit*) or (*bye*)

to quit clisp.

EOF (Ctrl-D)

to leave the current read-eval-print loop.

arrow keys

for editing and viewing the input history.

Tab key

to complete the symbol's name you are just typing.

Files

clisp

startup script

lisp.run

main executable

lispinit.mem

initial memory image

config.lsp

site-dependent configuration

- *.lisp
lisp source
- *.fas
lisp code, compiled by `clisp`
- *.lib
lisp source library information, generated and used by the `clisp` compiler
- *.c
C code, compiled from lisp source by `clisp`

Environment

CLISP_LANGUAGE

specifies the language `clisp` uses to communicate with the user. The value may be `english`, `deutsch`, `francais` and defaults to `english`. The `-L` option can be used to override this environment variable.

LC_CTYPE

specifies the locale which determines the character set in use. The value can be of the form *language* or *language_country* or *language_country.charset*, where *language* is a two-letter ISO 639 language code (lower case), *country* is a two-letter ISO 3166 country code (upper case). *charset* is an optional character set specification, and needs normally not be given because the character set can be inferred from the language and country.

LANG

specifies the language `clisp` uses to communicate with the user, unless it is already specified through the environment variable `CLISP_LANGUAGE` or the `-L` option. It also specifies the locale determining the character set in use, unless already specified through the environment variable `LC_CTYPE`. The value may begin with a two-letter ISO 639 language code, for example `en`, `de`, `fr`.

HOME and USER

are used for determining the value of the function `user-homedir-pathname`. (Unix implementation only.)

SHELL (Unix implementation only)

is used to find the interactive command interpreter called by `(shell)`.

TERM

determines the screen size recognized by the pretty printer. This environment variable is also mandatory for the built-in screen editor.

See also

[imprnotes.html](#), `cmucl(1)`, `xemacs(1)`.

Bugs

- The function `inspect` is not implemented.
- Not all extensions from CLtL2 are supported.
- No on-line documentation beyond `apropos` and `describe` is available.

Projects

- Writing on-line documentation.
- Write `inspect`.
- Enhance the compiler such that it can inline local functions.
- Specify a portable set of window and graphics operations.

Authors

Bruno Haible <haible@clisp.cons.org> and Michael Stoll.

CLISP manual page

Bruno Haible <haible@clisp.cons.org>

Last modified: 18 July 1999.