

NAME

mksls - update nelsis database from *.sls and *.fun sources

SYNOPSIS

mksls [-h] [-m] [topcell]

DESCRIPTION

Mksls searches the current project directory for *.sls and *.fun files that have been changed since the last time they were compiled. It then calls *csls* or *cfun*, whatever's appropriate, to compile a source file that is out of date. If *topcell* is specified, *mksls* calls *sls_exp* for this top level cell and then executes "sls topcell topcell.cmd". *Mksls* aborts if an error occurs in any of the subprocesses it spawns.

OPTIONS

- h Print a help screen, then exit.
- m Print a script on stdout that can be processed by *make(1)* to obtain the required recompilations, then exit.

RESTRICTIONS

The compiler *csls* requires sls source files to be compiled in a bottom up fashion. (You can get around this with the aid of the "external" statement, see the SIs User Manual.) Consequently, the order in which *csls* is called sometimes is important. *Mksls* has limited support for this restriction, see the entry "mk.slsfiles" in the FILES section below.

Mksls does not really know what a Nelsis database looks like. It stores timestamps of the compilation dates in its own private directory "mk.dates", see the entry "mk.dates" in the FILES section below. As a consequence, if you compile a file "by hand" --for instance by typing "csls myfile.sls"-- then the mk.dates directory does not get updated and *mksls* still assumes that "myfile.sls" needs a recompile. Fortunately, this never causes the database to be out of date, but of course it may waste a little cpu time. The bottom line: use *mksls* consistently.

FILES

The following files provide the user of *mksls* with a way to override *mksls*'s default actions and arguments for subprocesses. None of these files is obligatory. Basically you can forget about them.

- mk.slsfiles* This file contains the names of the files that need to be processed by *csls*. The order in which the filenames appear in *mk.slsfiles* is the order that *mksls* obeys while spawning subprocesses for recompilation. Default is "*.sls" (alphabetical order).
- mk.funfiles* This file contains the names of the files that need be processed by *cfun*. The order in which the filenames appear in *mk.funfiles* is the order that *mksls* obeys while spawning subprocesses for recompilation. Default is "*.fun".
- mk.topcell* Contains the name of the top level cell. Default is the cell name supplied on the command line, see the description above.
- mk.slsargs* Contains extra args for *csls*. Defaults to "-s".
- mk.funargs* Contains extra args for *cfun*. The default is "-s -Cg -CI\$INCLUDE",

where `$INCLUDE` is substituted with the contents of the file `mk.include` (see below).

`mk.expargs` Contains the argument to `sls_exp`.
Defaults to `"-g"`.

`mk.include` Contains directory names to be searched for include files. Include files must have a suffix `".h"`. The default is to have no include directories.

`mk.dates` This is the directory that `mksls` creates to keep track of the time a file was last processed. If you remove this directory --or entries in it-- *mksls* assumes that the missing timestamp is the beginning of the epoch, thus enforcing (re)compilation of that particular file.

ENVIRONMENT

Mksls inspects the following environment variables:

`SLS_MKDB` The program to call when a `*.sls` file is out of date. Default is `"csls"`.

`FUNC_MKDB` The program to call when a `*.fun` file is out of date. Default is `"cfun"`.
Commonly, this is set to `"dofunc"`, a front-end to `cfun` that enables debugging of the functional sources with a source-level debugger.

`SLS_EXP` The program called to expand the top level cell. Default is `"sls_exp"`.

SEE ALSO

`csls(1ICD)`, `cfun(1ICD)`, `make(1)`.

AUTHOR

Paul Stravers