NAME

selog − selective logging

SYNOPSIS

require "selog"

−− intialization
selog.open(*config*)
selog.open(*config*, {*name1*, *name2*, ...})

−− creating selectors
local *sel* = selog.new(*name*, *level*)
local *sel* = selog(*name*, *level*)
local *sel* = selog(*name*, "fatal(*status*)")
local *sel* = selog(*name*, "fatal", *status*)

−− inspecting selectors
local *name* = sel.name
local *level* = sel.level
local *status* = sel.exitval

if sel.on **then** ... **end**

−− logging
sel(*message...*)

DESCRIPTION

Selog is a library of routines that unifies error reporting, activity logging, and debug tracing. It allows programmers to give their users flexible control over which messages are written and where they are written to.

This manual Lua interface to selog by reference to the C manual, **selog**(3), and the end-user manual, **selog**(7).

Initialization

To configure and reconfigure selog, call **selog.open()** passing the configuration string as the first argument. The configuration syntax and other details are described in **selog**(7). If you want selog to check the category names in the configuration, you should pass an array-like table as the second argument that lists all the valid names. More details can be found in the "INITIALIZATION" section of **selog**(3).

Creating selectors

Logging is performed using selectors, which are created by calling **selog.new()** or just **selog()** itself. This takes two string arguments: the selector's category *name* and its *level*. More details can be found in the "SELECTORS" section of **selog**(3). The string equivalents for the levels are listed in **selog**(7). To specify the exit status of a "fatal" selector, you can either put it in round brackets at the end of the level string, or you can pass it as a third argument.

Inspecting selectors

You can extract information about a selector by treating it like a table. The available keys are:

exitval The exit status for "fatal" selectors, and **nil** for others.

level The canonical name of the selector's level.

name The category name given when the selector was created.

on A boolean value that is true if the selector is enabled. This can be used to disable message preparation code when a selector is disabled.

Logging

To log a message, just call the selector. The arguments are concatenated with spaces between. If the selector's level is "trace" then the message includes debug information about the caller.

SEE ALSO

selog(3), **selog**(7)

AUTHOR

Written by Tony Finch <dot@dotat.at> <fanf2@cam.ac.uk>
at the University of Cambridge Computing Service.

Source available from <http://dotat.at/prog/selog>