

lfescript(1)

Robert Virding

2013-2016

NAME

lfescript - Lisp Flavoured Erlang (LFE) scripting support

SYNOPSIS

lfescript provides support for running short LFE programs without having to compile them first and an easy way to retrieve the command line arguments.

EXPORTS

script-name script-arg1 script-arg2 ...

lfescript lfescript-flags script-name script-arg1 script-arg2 ...

lfescript runs a script written in LFE.

lfescript:script_name() -> File

Types:

File = filename()

The `script_name/0` function returns the name of the lfescript being executed. If the function is invoked outside the context of an lfescript, the behavior is undefined.

EXAMPLE

Here follows an example script.

```
$ cat factorial
#!/usr/bin/env lfescript
;; -*- mode: lfe -*-
;;; -smp enable -sname factorial -mnesia debug verbose

(defun main
  ([list string])
  (try
    (let* ((n (list_to_integer string))
          (f (fac n)))
      (lfe_io:format "factorial ~w = ~w\n" (list n f)))
    (catch
      ((tuple _ _ _) (usage))))
  ([_] (usage)))

(defun fac
  ([0] 1)
  ([n] (* n (fac (- n 1)))))

(defun usage ()
  (lfe_io:format "usage: factorial integer\n" ()))
```

The header of the LFE script is different from a normal LFE module. The first line is an interpreter line which invokes `lfescript` if the script is run as in the first command line above. If `lfescript` is explicitly invoked in the second command line above then this header line will be ignored. On the second or third line it is possible to give arguments to the emulator with the syntax:

```
;;! -smp enable -sname factorial -mnesia debug verbose
```

In the example the second line is an optional directive to Emacs which causes it to enter LFE mode when editing the script file.

The rest of the file contains LFE source code. It must always contain the function `main/1`. When the script is run this function will be called with a list of strings representing the arguments with which the script was called. It is possible to define, include and use macros in the source code.

The source code is checked and warnings and errors will be printed. If there are errors the script will not run and it will terminate with exit status 127. Otherwise the code will be interpreted. If the function `main/1` returns successfully then the exit status for the script will be 0 but if an exception is raised then exit status will be 127.

OPTIONS

The following option is accepted by `lfescript`

- `-s` - Only perform a syntactic and semantic check of the script file. Warnings and errors (if any) are written to the standard output, but the script will not be run. The exit status will be 0 if there were no errors, and 127 otherwise.

Unrecognised options are ignored.

ENVIRONMENT VARIABLES

LFESCRIPT_EMULATOR

The command used to start the emulator. Default is `'erl'`. This can be useful for passing arguments into the emulator, for example

```
LFESCRIPT_EMULATOR="erl -pa sune"
```

will add the directory `sune` to the code path.

SEE ALSO

`lfe(1)`, `lfe_guide(7)`