

**NAME**

lua-akfavatar – Lua interpreter for the AKFAvatar user interface

**SYNOPSIS**

**lua-akfavatar** [*options*] [*script* [*args* ]]

**DESCRIPTION**

The program lua-akfavatar is the Lua interpreter for the AKFAvatar user interface. It loads and executes Lua scripts in source form (not bytecode). Either Lua scripts can be loaded or ".avt" demo files. If no *script* is given, it starts with a file-selector.

The given *args* are available to *script* as strings in a global table named **arg**. The *script* itself is in **arg[0]**, the first argument is **arg[1]** and so on.

You have to use the command **local avt = require "lua-akfavatar"** to get access to the AKFAvatar specific functions.

With the file-selector you can also see documentation files with the file extension ".about". These can be simple text files encoded in UTF-8 or plain ASCII. They can use the "overstrike" technique for bold or underlined text. You can for example use the text-setting system "groff" to generate such files.

Furthermore GNU info files with the file extension ".info" can be shown as text (no support for hyperlinks).

**OPTIONS**

- h
- help   print a short help-message
- v
- version
- print version information
- l [*var*=]*name*
- call **require("name")** before executing the script. Sets a global variable with the *name* or *var*. Typically used to load libraries.
- f
- fullscreen
- start in fullscreen mode (only if no script is given)
- F
- Fullscreen
- start in fullscreen mode without switching the resolution (only if no script is given)
- dir=*dir*
- start the file-selector in that directory (only if no script is given)

**ENVIRONMENT**

LUA\_PATH\_5\_2

LUA\_PATH

searchpath for Lua modules. Templates are separated with semicolons. A question mark (?) is replaced with the module name. A dot in the module name is replaced with the directory separator. Two semicolons (;;) add the default path.

LUA\_CPATH\_5\_2

LUA\_CPATH

searchpath for C modules. Syntax like in *LUA\_PATH*.

AVTDATAPATH

searchpath for data files (images, audio, texts...). It gets stored in the variable **avt.datapath** and can be used with the function **avt.search(filename)**. The directories are separated with semicolons. Unlike the former paths there are no templates with question marks.

**EXAMPLE**

```
local avt = require "lua-akfavatar"
avt.encoding("UTF-8")
avt.title("my first program")
avt.start()
avt.avatar_image("default")
avt.say("Hello there!\n")
local name = avt.ask("What's your name? ")
avt.say("Hello, ", name, ". Nice to meet you!\n")
avt.wait_button()
```

**SECURITY**

Lua is a complete programming language. There are no sandbox limitations. Any such programming language can be misused to write malicious code. So keep in mind that Lua scripts are programs and should be treated with the same care as other programs. Some other input files for Lua-AKFAvatar could contain code, too.

**AUTHORS**

AKFAvatar

Lua-AKFAvatar    Andreas K. Foerster

Lua                R. Ierusalimschy, L. H. de Figueiredo, and W. Celes

**HOME PAGE**

<http://akfavatar.nongnu.org/>

**SEE ALSO**

**lua(1)** **lua-akfavatar-ref(3)** **akfavatar-graphic(3)** **akfavatar-term(3)** **akfavatar.utf8(3)**

<http://akfavatar.nongnu.org/manual/>

<http://www.lua.org/manual/5.2/>