

# The lualibs package

Elie Roux

`elie.roux@telecom-bretagne.eu`

2010/05/10 v0.93

## Abstract

Additional lua functions taken from the libs of ConT<sub>E</sub>Xt. For an introduction on this package (among others), please refer to the document `luatex-reference.pdf`.

## 1 Overview

Lua is a very minimal language, and it does not have a lot of built-in functions. Some functions will certainly be needed by a lot of packages. Instead of making each of them implement these functions, the aim of this package is to provide a standard set of functions. All functions are taken from ConT<sub>E</sub>Xt libraries.

There are some differences with the ConT<sub>E</sub>Xt functions though, especially on names: for example the `file.*` functions are renamed in `fpath.*`. It seems more logical as they deal with file paths, not files. Also the `file.is_readable` and `file.is_writable` are renamed `lfs.is_readable` and `lfs.is_writable`.

If you use a function you think is missing in this file, please tell the maintainer.

**Warning:** Even if the names will certainly remain the same, some implementations may differ, and some functions might appear or disappear. As LuaT<sub>E</sub>X is not stable, this file is not neither.

All functions are described in this document, but the one of the functions you'll use most will certainly be `table.serialize` (also named `table.tostring`) that takes a table and returns an intended string describing the table. It describes the table so that LuaT<sub>E</sub>X can read it again as a table. You can do a lot of things with this functions, like printing a table for debugging, or saving a table into a file. Functions are also converted into bytecode to be saved.

## 2 lualibs.lua

```
1 do
2   local lualibs_module = {
3     name      = "lualibs",
4     version   = 0.93,
5     date      = "2010/05/10",
```

```

6      description = "Lua additional functions.",
7      author      = "Hans Hagen, PRAGMA-ADE, Hasselt NL & Elie Roux",
8      copyright   = "PRAGMA ADE / ConTeXt Development Team",
9      license      = "See ConTeXt's mreadme.pdf for the license",
10   }
11   if luatexbase then
12       luatexbase.provides_module(lualibs_module)
13   end
14 end

```

Initialize Kpathsea library, so that `require()` will use it to locate modules.

```

15
16 kpse.set_program_name("luatex")
17

```

We could load the modules with `require`, but this works only from version 0.44, and doesn't work in Debian, so we simply use `dofile`.

```

18
19 local function load_lualibs_module(filename)
20     local path = kpse.find_file(filename)
21     if not path then
22         texio.write_nl(string.format("lualibs: error: cannot find file %s", filename))
23         return
24     end
25     texio.write_nl('log', string.format("lualibs: loading file %s", path))
26     --if (tex and tex.luatexversion and tex.luatexversion > 44)
27     --    or (status and status.luatex_version and status.luatex_version > 44) then
28     --    require(filename)
29     --else
30         dofile(path)
31     --end
32 end
33
34 load_lualibs_module("lualibs-string.lua")
35 load_lualibs_module("lualibs-lpeg.lua")
36 load_lualibs_module("lualibs-boolean.lua")
37 load_lualibs_module("lualibs-number.lua")
38 load_lualibs_module("lualibs-math.lua")
39 load_lualibs_module("lualibs-table.lua")
40 load_lualibs_module("lualibs-aux.lua")
41 load_lualibs_module("lualibs-io.lua")
42 load_lualibs_module("lualibs-os.lua")
43 load_lualibs_module("lualibs-file.lua")
44 load_lualibs_module("lualibs-md5.lua")
45 load_lualibs_module("lualibs-dir.lua")
46 load_lualibs_module("lualibs-unicode.lua")
47 load_lualibs_module("lualibs-utils.lua")
48 load_lualibs_module("lualibs-dimen.lua")
49 load_lualibs_module("lualibs-url.lua")
50 load_lualibs_module("lualibs-set.lua")
51 load_lualibs_module("lualibs-dimen.lua")

```

52

Aliases for backward compatibility.

53

54 `fpath` = `file`

55 `fpath.split` = `file.split_path`

56 `lfs.is_readable` = `file.is_readable`

57 `lfs.is_writable` = `file.is_writable`

58

Functions no longer provided by ConTeXt module.

`string:stripspaces` A function to strip the spaces at the beginning and at the end of a string.

59

60 `function string:stripspaces()`

61     `return (self:gsub("^%s*(.*)%s*$", "%1"))`

62 `end`

63

`lpeg.space` and `lpeg.newline` Two small helpers for `lpeg`, that will certainly be widely used: spaces and newlines.

64

65 `lpeg.space` = `lpeg.S(" \t\f\v")`

66 `lpeg.newline` = `lpeg.P("\r\n") + lpeg.P("\r") + lpeg.P("\n")`

67

`table.contains value` A function returning true if the value `val` is in the table `t`.

68

69 `function table.contains_value(t, val)`

70     `if t then`

71         `for k, v in pairs(t) do`

72             `if v==val then`

73                 `return true`

74             `end`

75         `end`

76     `end`

77     `return false`

78 `end`

79

`table.contains key` A function returning true if the key `key` is in the table `t`

80

81 `function table.contains_key(t, key)`

82     `if t then`

83         `for k, v in pairs(t) do`

84             `if k==key then`

85                 `return true`

86             `end`

87         `end`

88     `end`

89     `return false`

90 `end`

91

`table.value position` A function returning the position of a value in a table. This will be important to be able to remove a value.

```
92
93 function table.value_position(t, val)
94     if t then
95         local i=1
96         for k, v in pairs(t) do
97             if v==val then
98                 return i
99             end
100             i=i+1
101         end
102     end
103     return 0
104 end
105
```

`table.key position` A function returning the position of a key in a table.

```
106
107 function table.key_position(t, key)
108     if t then
109         local i=1
110         for k,v in pairs(t) do
111             if k==key then
112                 return i
113             end
114             i = i+1
115         end
116     end
117     return -1
118 end
119
```

`table.remove key` Removing a key from a table.

```
120
121 function table.remove_key(t, k)
122     local p = table.key_position(t,k)
123     if p ~= -1 then
124         table.remove(t, table.key_position(t,k))
125     end
126 end
127
```

`fpath.normalize sep` A function to change directory separators to canonical ones (/).

```
128
129 function fpath.normalize_sep(str)
130     return str:gsub("\\", "/")
131 end
132
```

`fpath.localize_sep` A function changing directory separators into local ones (/ on Unix, \ on Windows).

```
133
134 function fpath.localize_sep(str)
135     if os.type == 'windows' or os.type == 'msdos' then
136         return str:gsub("/", "\\")
137     else
138         return str:gsub("\\", "/")
139     end
140 end
141
```