

The luatexbase-compatible package

Heiko Oberdiek (primary author of luatex)
Élie Roux, Manuel Pégourié-Gonnard, Philipp Gesang*

<https://github.com/lualatex/luatexbase>
luatex-dev@tug.org

2011/05/24 v0.4

Abstract

The Lua_{TeX} manual is very clear: everything may change. This package provides tools to help package writers deal with the changes. It helps supporting Lua_{TeX} versions down to 0.25.4, and is tested with Lua_{TeX} 0.40.6 (T_{EX} Live 2009), Lua_{TeX} 0.60.2 (T_{EX} Live 2010) and current beta versions.

The supported formats are Plain and L^AT_{EX} adapted for Lua_{TeX} as provided by T_{EX} Live and MiK_{TeX} (see `luatex-doc.pdf` section 4 for details about these formats).

Contents

1	Documentation	1
2	Implementation	2
2.1	Preliminaries	2
2.2	<code>\directlua abstraction</code>	4
2.3	Version information	4
2.4	Primitives	4
3	Test files	5

1 Documentation

Three problems are currently addressed by this package: changes in the syntax of `\directlua`, version information, and variable policies for primitives activation and naming (in Lua_{TeX} itself as well as in the formats provided by distributions).

Older versions of Lua_{TeX} used to support multiple Lua states. A number was mandatory with `\directlua` in order to specify the Lua state to be used. Later, support for multiple Lua states was removed and the old syntax resulted in a warning. Now (Lua_{TeX} 0.50), `\directlua` again accepts a number after `\directlua`, but with a different meaning (see the Lua_{TeX} manual for details).

*See “History” in [luatexbase.pdf](#) for details.

This package provides a macro `\luatexbase@directlua` that expands to `\directlua0` on LuaTeX 0.35 and lower (where the number is mandatory), and to `\directlua` otherwise. It is a macro in both cases so that it always expands in exactly two steps.

Current versions of LuaTeX make the version available directly from Lua as `tex.luatexversion` and `tex.luatexrevision`. However, older versions (such as 0.25.4) didn't, which makes it particularly uneasy to test the version from within Lua. The present package makes this information available as `luatexbase.luatexversion` and `luatexbase.luatexrevision`.

Starting with LuaTeX 0.39.0, the only primitives available in IniTeX mode are the basic primitives from TeX82 and `\directlua`. All other primitives are hidden by default and have to be activated using a Lua function. In TeX Live 2009 (LuaTeX 0.40.6), the following arrangement has been made in order to try preserving usability while avoiding name clashes in the L^ATeX world: in L^ATeX-based formats, all pdfTeX primitives are enabled with their normal name, but the primitives specific to LuaTeX are enabled with the `luatex` prefix.¹ In Plain based formats however, all the primitives are enabled with their natural name, but are also provided with the same name as in L^ATeX-based formats in order to help writing generic packages.

So, starting with TeX Live 2009, the situation is clear: the prefixed version of the LuaTeX primitives is always available. But in earlier versions (TeX Live 2008, LuaTeX 0.25.4) those primitives were available only with their natural names. The primitives provided by ϵ -TeX and pdfTeX on the other hand, are always available.

`\luatexbase@ensure@primitive{<name>}`

The tool provided to deal with that is `\luatexbase@ensure@primitive`, whose argument is a LuaTeX primitive name (without a leading backslash nor any `luatex` prefix, eg just `{latelua}`). It makes sure that the primitive gets available as `\luatex<name>`.

Warning. In particular circumstances, this macro may fail silently for primitives whose natural name starts with `luatex`, hence such primitives shouldn't be used as arguments. This is actually not a problem, since the only three such primitives are `\luatexversion`, `\luatexrevision` and `\luatexdatestamp`. The first two are already activated by `ifluatex` which is loaded by this package, so you don't need to activate them yourself. The third should never be used in production according to the LuaTeX manual.

Remark. If you only aim at compatibility down to TeX Live 2009 (LuaTeX 0.40.6), then you can simply use the primitives with their prefixed name (except for `\directlua` which never needs a prefix). If you want extra security and/or compatibility down to TeX Live 2008 (LuaTeX 0.25.4) then you should use `\luatexbase@ensure@primitive` for each primitive you intend to use (except `\directlua` again).

This package doesn't try to activate every primitive, since it would require an extensive list of primitives for each version of LuaTeX, so it seems simpler to leave that burden on package writers.

2 Implementation

1 `{*texpackage}`

2.1 Preliminaries

Catcode defenses and reload protection.

¹The prefix is dropped for primitives whose name already starts with `luatex`.

```

2 \begingroup\catcode61\catcode48\catcode32=10\relax% = and space
3 \catcode123 1 % {
4 \catcode125 2 % }
5 \catcode 35 6 % #
6 \toks0\expandafter{\expandafter\endlinechar\the\endlinechar}%
7 \edef\x{\endlinechar13}%
8 \def\y#1 #2 {%
9 \toks0\expandafter{\the\toks0 \catcode#1 \the\catcode#1}%
10 \edef\x{\x \catcode#1 #2}}%
11 \y 13 5 % carriage return
12 \y 61 12 % =
13 \y 32 10 % space
14 \y 123 1 % {
15 \y 125 2 % }
16 \y 35 6 % #
17 \y 64 11 % @ (letter)
18 \y 10 12 % new line ^^J
19 \y 39 12 % '
20 \y 40 12 % (
21 \y 41 12 % )
22 \y 42 12 % *
23 \y 44 12 % ,
24 \y 45 12 % -
25 \y 46 12 % .
26 \y 47 12 % /
27 \y 58 12 % :
28 \y 60 12 % <
29 \y 91 12 % [
30 \y 93 12 % ]
31 \y 94 7 % ^
32 \y 96 12 % `
33 \toks0\expandafter{\the\toks0 \relax\noexpand\endinput}%
34 \edef\y#1{\noexpand\expandafter\endgroup%
35 \noexpand\ifx#1\relax \edef#1{\the\toks0}\x\relax%
36 \noexpand\else \noexpand\expandafter\noexpand\endinput%
37 \noexpand\fi}%
38 \expandafter\y\csname luatexbase@compat@sty@endinput\endcsname%
Package declaration.
39 \begingroup
40 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
41 \def\x#1[#2]{\immediate\write16{Package: #1 #2}}
42 \else
43 \let\x\ProvidesPackage
44 \fi
45 \expandafter\endgroup
46 \x{luatexbase-compat}[2011/05/24 v0.4 Compatibility tools for LuaTeX]
Make sure LuaTeX is used.
47 \begingroup\expandafter\expandafter\expandafter\endgroup
48 \expandafter\ifx\csname RequirePackage\endcsname\relax
49 \input ifluatex.sty
50 \else
51 \RequirePackage{ifluatex}

```

```

52 \fi
53 \ifluatex\else
54 \begingroup
55 \expandafter\ifx\csname PackageError\endcsname\relax
56 \def\x#1#2#3{\begingroup \newlinechar10
57 \errhelp{#3}\errmessage{Package #1 error: #2}\endgroup}
58 \else
59 \let\x\PackageError
60 \fi
61 \expandafter\endgroup
62 \x{luatexbase-compat}{LuaTeX is required for this package. Aborting.}{%
63 This package can only be used with the LuaTeX engine^^J%
64 (command 'lualatex' or 'luatex').^^J%
65 Package loading has been stopped to prevent additional errors.}
66 \expandafter\luatexbase@compat@sty@endinput%
67 \fi

```

2.2 \directlua abstraction

Define `\luatexbase@directlua` to be either `\directlua0` or `\directlua`, depending on the version of LuaTeX.

```

68 \begingroup
69 \expandafter\ifx\csname newcommand\endcsname\relax
70 \toks0{\long\def\luatexbase@directlua}%
71 \else
72 \toks0{\newcommand\luatexbase@directlua}%
73 \fi
74 \ifnum\luatexversion<36
75 \toks0\expandafter{\the\toks0{\directlua0}}%
76 \else
77 \toks0\expandafter{\the\toks0{\directlua}}%
78 \fi
79 \expandafter\endgroup
80 \the\toks0

```

2.3 Version information

Make `\luatexversion` and `\luatexrevision` available from Lua.

```

81 \luatexbase@directlua{%
82 luatexbase = luatexbase or {}
83 luatexbase.luatexversion = \number\luatexversion\space
84 luatexbase.luatexrevision = \number\luatexrevision\space}

```

2.4 Primitives

Try reasonably hard to activate a primitive. First, check if it is already activated and do nothing in this case.

```

85 \begingroup
86 \expandafter\ifx\csname newcommand\endcsname\relax
87 \toks0{\def\luatexbase@ensure@primitive#1}
88 \else

```

```

89 \toks0{\newcommand*\luatexbase@ensure@primitive[1]}
90 \fi
91 \toks2{}\def\x#1{\toks2\expandafter{\the\toks2 #1}}
92 \x{%
93 \ifcsname luatex#1\endcsname \else}
94 \ifnum\luatexversion<37\relax

```

tex.enableprimitives() not available. If the unprefixd primitive is undefined, issue an error.

```

95 \x{%
96 \begingroup\expandafter\expandafter\expandafter\endgroup
97 \expandafter\ifx\csname #1\endcsname\relax}
98 \begingroup\expandafter\expandafter\expandafter\endgroup
99 \expandafter\ifx\csname PackageError\endcsname\relax
100 \x{%
101 \errmessage{%
102 Package luatexbase-compat error: failed to enable '#1'.}}
103 \else
104 \x{%
105 \PackageError{luatexbase-compat}{%
106 Package luatexbase-compat error: failed to enable '#1'.}}{}
107 \fi
108 \x{%
109 \else}

```

Use the unprefixd primitive to define the prefixed version.

```

110 \x{%
111 \expandafter\let\csname luatex#1\expandafter\endcsname
112 \csname#1\endcsname
113 \fi}
114 \else

```

tex.enableprimitives() available, use it.

```

115 \x{%
116 \luatexbase@directlua{tex.enableprimitives('luatex', {'#1'})}}
117 \fi
118 \x{%
119 \fi}
120 \toks0\expandafter{\the\toks0\expandafter{\the\toks2}}
121 \expandafter\endgroup
122 \the\toks0

```

That's all folks!

```

123 \luatexbase@compat@sty@endinput%
124 /</texpackage>

```

3 Test files

Test files for Plain and LaTeX

```

125 <testplain>\input luatexbase-compat.sty
126 <testlatex>\RequirePackage{luatexbase-compat}
127 <*testplain, testlatex>

```

```
128 \catcode64 11
129 \luatexbase@directlua{local answer = 42}
130 \luatexbase@ensure@primitive{primitive}
131 \luatexprimitive\relax
132 \luatexbase@directlua{assert(type(luatexbase.luatexversion) == 'number')}
133 \luatexbase@directlua{assert(type(luatexbase.luatexrevision) == 'number')}
134 </testplain, testlatex>
135 (testplain)\bye
136 (testlatex)\stop
```