The luacolor package

Heiko Oberdiek*

2021-02-17 v1.17

Abstract

Package ${\sf luacolor}$ implements color support based on LuaTeX's node attributes.

Contents

1	Dog	cumentation 2
	1.1	Introduction
	1.2	Usage
	1.3	Limitations
2	Imp	plementation 3
	2.1	Catcodes and identification
	2.2	Check for LuaT _F X
	2.3	Check for disabled colors
	2.4	Load module and check version
	2.5	Find driver
	2.6	Attribute setting
	2.7	Whatsit insertion
	2.8	\pdfxform/\saveboxresource support 6
	2.9	Lua module
		2.9.1 Driver detection
		2.9.2 Color strings
		2.9.3 Attribute register
		2.9.4 Whatsit insertion
3	Inst	callation 12
	3.1	Download
	3.2	Bundle installation
	3.3	Package installation
	3.4	Refresh file name databases
	3.5	Some details for the interested
4	His	tory 13
		7/12/12 v1.0]
		9/04/10 v1.1
		0/03/09 v1.2
		$\frac{0}{12}$ v1.3
		$\frac{1}{03}$ /29 v1.4
		$\frac{1}{04/22}$ v1.5]
	L	

^{*}Please report any issues at https://github.com/ho-tex/luacolor/issues

5	Index	15
	[2021-02-17 v1.17]	15
	[2020-04-04 v1.16]	15
	[2020-02-24 v1.15]	15
	[2020-02-22 v1.14]	15
	[2019/11/29 v1.13]	15
	[2019/07/25 v1.12]	15
	[2018/11/22 v1.11]	14
	[2016/05/16 v1.10]	14
	[2016/05/13 v1.9]	14
	[2011/11/01 v1.8]	14
	[2011/10/22 v1.7]	14
	[2011/04/23 v1.6]	14

1 Documentation

1.1 Introduction

This package uses a LuaTEX's attribute register to to annotate nodes with color information. If a color is set, then the attribute register is set to this color and all nodes created in its scope (current group) are annotated with this attribute. Now the color property behaves much the same way as the font property.

1.2 Usage

Package color is loaded automatically by this package luacolor. If you need a special driver option or you prefer package xcolor, then load it before package luacolor, for example:

\usepackage[dvipdfmx]{xcolor}

The package luacolor is loaded without options:

\usepackage{luacolor}

It is able to detect PDF mode and DVI drivers are differentiated by its color specials. Therefore the package do need driver options.

Then it redefines the color setting commands to set attributes instead of whatsits for color.

At last the attribute annotations of the nodes in the output box must be analyzed to insert the necessary color whatsits. Currently LuaTeX lacks an appropriate callback function. Therefore package atbegshi is used to get control before a box is shipped out.

\luacolorProcessBox $\{\langle box \rangle\}$

Macro \luacolorProcessBox processes the box $\langle box \rangle$ in the previously described manner. It is automatically called for pages, but not for XForm objects. Before passing a box to \pdfxform, call \luacolorProcessBox first.

1.3 Limitations

Ligatures with different colored components: Package luacolor sees the ligature after the paragraph building and page breaking, when a page is to be shipped out. Therefore it cannot break ligatures, because the components might occupy different space. Therefore it is the respondibility of the ligature forming process to deal with different colored glyphs that form a ligature. The user can avoid the problem entirely by explicitly breaking the ligature at the places where the color changes.

. . .

2 Implementation

1 (*package)

2.1 Catcodes and identification

```
2 \begingroup\catcode61\catcode48\catcode32=10\relax%
    \catcode13=5 % ^^M
    \endlinechar=13 %
    \catcode123=1 % {
    \catcode125=2 % }
    \catcode64=11 % @
    \def\x{\endgroup
9
      \expandafter\edef\csname LuaCol@AtEnd\endcsname{%
10
        \endlinechar=\the\endlinechar\relax
11
        \catcode13=\the\catcode13\relax
12
        \catcode32=\the\catcode32\relax
        \catcode35=\the\catcode35\relax
13
        \catcode61=\the\catcode61\relax
14
        \catcode64=\the\catcode64\relax
15
        \catcode123=\the\catcode123\relax
16
        \catcode125=\the\catcode125\relax
17
      }%
18
    }%
19
20 \x\catcode61\catcode48\catcode32=10\relax%
21 \catcode13=5 % ^^M
22 \endlinechar=13 %
23 \catcode35=6 % #
24 \catcode64=11 % @
25 \catcode123=1 % {
26 \catcode125=2 % }
27 \def\TMP@EnsureCode#1#2{%
    \edef\LuaCol@AtEnd{%
      \LuaCol@AtEnd
29
      \catcode#1=\the\catcode#1\relax
30
    }%
31
32
    \catcode#1=#2\relax
33 }
34 \TMP@EnsureCode{34}{12}% "
35 \TMP@EnsureCode{39}{12}%,
36 \TMP@EnsureCode{40}{12}% (
37 \TMP@EnsureCode{41}{12}%)
38 \TMP@EnsureCode{42}{12}% *
39 \TMP@EnsureCode{43}{12}% +
40 \TMP@EnsureCode{44}{12}%,
41 \TMP@EnsureCode{45}{12}% -
```

```
42 \TMP@EnsureCode{46}{12}% .

43 \TMP@EnsureCode{47}{12}% /

44 \TMP@EnsureCode{58}{12}% :

45 \TMP@EnsureCode{60}{12}% <

46 \TMP@EnsureCode{62}{12}% >

47 \TMP@EnsureCode{91}{12}% [

48 \TMP@EnsureCode{93}{12}% ]

49 \TMP@EnsureCode{95}{12}% _ (other!)

50 \TMP@EnsureCode{96}{12}% '

51 \edef\LuaCol@AtEnd{\LuaCol@AtEnd\noexpand\endinput}

Package identification.

52 \NeedsTeXFormat{LaTeX2e}

53 \ProvidesPackage{luacolor}%

54 [2021-02-17 v1.17 Color support via LuaTeX's attributes (HO)]
```

2.2 Check for LuaT_EX

Without LuaTEX there is no point in using this package.

```
55 \RequirePackage{color}

56 \ifx\directlua\@undefined

57 \PackageError{luacolor}{%

58 This package may only be run using LuaTeX%

59 }\@ehc

60 \expandafter\LuaCol@AtEnd

61 \fi%
```

2.3 Check for disabled colors

```
62 \ifcolors@
63 \ensuremath{\setminus} else
    \PackageWarningNoLine{luacolor}{%
64
      Colors are disabled by option 'monochrome'%
65
66
    }%
    \def\set@color{}%
    \def\reset@color{}%
68
    \def\set@page@color{}%
69
    \def\define@color#1#2{}%
    \expandafter\LuaCol@AtEnd
71
72 \fi%
```

2.4 Load module and check version

```
73 \directlua{%
   require("luacolor")%
74
75 }
76 \begingroup
    \edef\x{\directlua{tex.write("2021-02-17 v1.17")}}%
77
78
    \edef\y{%
      \directlua{%
79
        if oberdiek.luacolor.getversion then %
80
           oberdiek.luacolor.getversion()%
81
        end%
82
83
      }%
   }%
84
    \int x x y
85
86
    \else
87
      \PackageError{luacolor}{%
88
        Wrong version of lua module.\MessageBreak
```

```
Package version: \x\MessageBreak
                                                          90
                                                                                 Lua module: \y
                                                          91
                                                                           }\@ehc
                                                                      \fi
                                                          92
                                                          93 \endgroup
                                                       2.5
                                                                         Find driver
                                                          94 \in \output mode = \output mode
                                                          95 \ensuremath{\setminus} \texttt{else}
                                                          96
                                                                      \begingroup
                                                          97
                                                                            \def\current@color{}%
                                                          98
                                                                            \def\reset@color{}%
                                                          99
                                                                            \sc \z@=\hbox{%}
                                                        100
                                                                                  \begingroup
                                                        101
                                                                                        \set@color
                                                                                  \endgroup
                                                        102
                                                       103
                                                                            \edef\reserved@a{%
                                                       104
                                                                                  \directlua{%
                                                       105
                                                                                        oberdiek.luacolor.dvidetect()%
                                                       106
                                                       107
                                                                            }%
                                                       108
                                                       109
                                                                            \ifx\reserved@a\@empty
                                                                                  \PackageError{luacolor}{%
                                                       110
                                                                                        DVI driver detection failed because of\MessageBreak
                                                       111
                                                       112
                                                                                        unrecognized color \string\special
                                                       113
                                                                                 \ \ensuremath{\mbox{Qehc}}
                                                       114
                                                                                  \endgroup
                                                       115
                                                                                  \expandafter\expandafter\LuaCol@AtEnd
                                                       116
                                                                                  \PackageInfo{luacolor}{%
                                                       117
                                                                                       Type of color \string\special: \reserved@a
                                                       118
                                                                                  \@gobble}%
                                                       119
                                                                            \fi%
                                                       120
                                                       121
                                                                      \endgroup
                                                       122 \fi
                                                       2.6
                                                                          Attribute setting
\LuaCol@Attribute
                                                       123 \newattribute\LuaCol@Attribute
                                                       125 \directlua{%
                                                       oberdiek.luacolor.setattribute(\number\allocationnumber)%
                                                       127 }
                    \set@color
                                                       128 \protected\def\set@color{%
                                                                      \LuaCol@setattribute\LuaCol@Attribute{%
                                                       129
                                                                            \directlua{%
                                                       130
                                                                                  oberdiek.luacolor.get("\luaescapestring{\current@color}")%
                                                       131
                                                       132
                                                                           }%
                                                       133
                                                                     }%
                                                       134 }
               \reset@color
                                                       135 \def\reset@color{}
```

89

2.7 Whatsit insertion

\luacolorProcessBox

```
136 \def\luacolorProcessBox#1{%
137
     \directlua{%
       oberdiek.luacolor.process(\number#1)%
138
139
     }%
140 }
141 \directlua{%
142
     if luatexbase.callbacktypes.pre_shipout_filter then
       token.get_next()
143
144
     end
145 }\@secondoftwo\@gobble{
     \RequirePackage{atbegshi}[2011/01/30]
147
     \AtBeginShipout{%
148
       \luacolorProcessBox\AtBeginShipoutBox
149
150 }
   Set default color.
151 \set@color
```

2.8 \pdfxform/\saveboxresource support

```
152 \ifnum\outputmode=\One
153 \let\LuaColOorgOpdfxform\saveboxresource
```

First we need some helpers to allow expandable code to parse keyword style arguments:

```
\def\LuaCol@iii@i@ii#1#2#3{#3{#1}{#2}}
154
155
       \def\LuaCol@ii@i#1#2{{#2#1}}
       \def\LuaCol@if@keyword#1#2#3{%
156
157
         \expanded{\unexpanded{\LuaCol@iii@i@ii{#2}{#3}}\expandafter}%
         \directlua{%
158
159
           token.put_next(token.create(token.scan_keyword(token.scan_string()))
160
           and '@firstoftwo'
           or '@secondoftwo'))
161
         }{#1}%
162
       }
163
```

The following macro scans a integer and expands to a token equivalent to a chardef whose value corresponds to the scanned integer. This allows the integer to be passed around as a undelimited argument.

```
164 \def\LuaCol@scan@number{%
165 \directlua{
166    token.put_next(token.new(token.scan_int(), token.command_id'char_given'))
167   }%
168 }
```

TEX primitives like \saveboxresource read braced arguments in a special way. Especially they expand everything until they find a left brace. To simulate this, we use Lua to expand everything else:

```
176
           token.put_next(t)
177
         }%
       }
178
179
       \def\LuaCol@scan@boxresource@i#1#2{%
         \LuaCol@if@keyword{attr}{%
180
           \expanded{\unexpanded{\LuaCol@scan@boxresource@iI{#1#2attr}}%
181
              \expandafter\expandafter\expandafter}%
182
           \LuaCol@scan@tobrace
183
         }{%
184
           \LuaCol@scan@boxresource@ii{#1#2}%
185
         }%
186
187
       \def\LuaCol@scan@boxresource@iI#1#2{\LuaCol@scan@boxresource@ii{#1{#2}}}
188
       \def\LuaCol@scan@boxresource@ii#1{%
189
         \LuaCol@if@keyword{resources}{%
190
191
           \expanded{\unexpanded{\LuaCol@scan@boxresource@iiI{#1resources}}%
192
              \expandafter\expandafter\expandafter}%
           \LuaCol@scan@tobrace
193
194
           \LuaCol@scan@boxresource@iii{#1}%
195
         }%
196
197
       \def\LuaCol@scan@boxresource@iiI#1#2{\LuaCol@scan@boxresource@iii{#1{#2}}}
198
       \def\LuaCol@scan@boxresource@iii#1{%
199
         \LuaCol@if@keyword{margin}{%
200
201
           \expanded{\unexpanded{\LuaCol@scan@boxresource@iv{#1margin }}%
202
              \expandafter\expandafter\expandafter}%
           \LuaCol@scan@number
203
204
         }{%
205
           \LuaCol@scan@boxresource@iv{#1}{}%
         }%
206
       }
207
       \def\LuaCol@scan@boxresource@iv#1#2{%
208
         \expanded{\unexpanded{\LuaCol@scan@boxresource@v{#1#2}}%
209
           \expandafter\expandafter\expandafter}%
210
         \LuaCol@scan@number
211
       }
212
       \def\LuaCol@scan@boxresource@v#1#2{%
213
214
         \luacolorProcessBox{#2}%
215
         \LuaCol@org@pdfxform#1#2%
216
217
```

This could be written in Lua, but at least upto LuaTEX 1.11, feeding back too many tokens from Lua to TEX triggers a segmentation fault. This is written in Lua so the integer setting is expandable and does not interfere with a preceding \immediate.

```
\protected\def\saveboxresource{%
218
         \LuaCol@if@keyword{type}{%
219
            \expandafter
220
            \expanded{\unexpanded{\LuaCol@scan@boxresource@i{type }}%
221
              \expandafter\expandafter\expandafter}%
222
223
           \LuaCol@scan@number
         }{%
            \LuaCol@scan@boxresource@i{}{}%
225
226
         }%
       }
227
```

Legacy alias.

```
228
                                                                 \let\pdfxform\saveboxresource
                                           229 \fi
                                           230 \LuaCol@AtEnd%
                                           231 (/package)
                                           2.9 Lua module
                                           232 (*lua)
                                           Box zero contains a \hbox with the color \special. That is analyzed to get the
                                           prefix for the color setting \special.
                                           233 oberdiek = oberdiek or {}
                                           234 local luacolor = oberdiek.luacolor or {}
                                           235 oberdiek.luacolor = luacolor
getversion()
                                           236 function luacolor.getversion()
                                           237 tex.write("2021-02-17 v1.17")
                                           238 end
                                           2.9.1 Driver detection
                                           239 local ifpdf = tonumber(tex.outputmode or tex.pdfoutput) > 0
                                           240 local prefix
                                           241 local prefixes = {
                                           242 dvips = "color",
                                           243 dvipdfm = "pdf:sc "
                                           244 truetex = "textcolor:",
                                          245 pctexps = "ps::",
                                          246 }
                                          247 \log 247 
                                          248 ["^color "]
                                                                                                                                = "dvips",
                                           249 ["^pdf: *begincolor "] = "dvipdfm",
                                           250 ["^pdf: *bcolor "]
                                                                                                                            = "dvipdfm",
                                           251 ["^pdf: *bc "]
                                                                                                                               = "dvipdfm",
                                           252 ["^pdf: *setcolor "] = "dvipdfm",
                                           253 ["^pdf: *scolor "]
                                                                                                                               = "dvipdfm",
                                           254 ["^pdf: *sc "]
                                                                                                                               = "dvipdfm",
                                           255 ["^textcolor:"]
                                                                                                                               = "truetex",
                                           256
                                                          ["^ps::"]
                                                                                                                                 = "pctexps",
                                           257 }
                  info()
                                           258 local function info(msg, term)
                                           259 local target = "log"
                                                        if term then
                                           260
                                           261
                                                                target = "term and log"
                                           262
                                           263 texio.write_nl(target, "Package luacolor info: " .. msg .. ".")
                                           264 texio.write_nl(target, "")
                                           265 end
  dvidetect()
                                           266 function luacolor.dvidetect()
                                           267 local v = tex.box[0]
                                                        assert(v.id == node.id("hlist"))
                                                       for v in node.traverse_id(node.id("whatsit"), v.head) do
                                           269
                                           270
                                                               if v and v.subtype == node.subtype("special") then
                                                                      local data = v.data
                                           271
                                                                      for pattern, driver in pairs(patterns) do
                                           272
```

```
273
                            if string.find(data, pattern) then
                274
                              prefix = prefixes[driver]
                275
                              tex.write(driver)
                276
                              return
                 277
                            end
                278
                          end
                          info("\\special{" .. data .. "}", true)
                279
                          return
                280
                281
                        end
                282 end
                 283 info("Missing \\special", true)
                 284 \; \mathrm{end}
                2.9.2 Color strings
                 285 \log 1 map = {
                 286 \quad n = 0,
                 287 }
         get()
                288 function luacolor.get(color)
                289 tex.write("" .. luacolor.getvalue(color))
                 290 end
    getvalue()
                 291 function luacolor.getvalue(color)
                292 local n = map[color]
                293 if not n then
                        n = map.n + 1
                294
                        map.n = n
                295
                        map[n] = color
                296
                 297
                        map[color] = n
                 298
                      end
                     return n
                 299
                 300 end
                 2.9.3 Attribute register
setattribute()
                301 local attribute
                 302 function luacolor.setattribute(attr)
                303 attribute = attr
                304 \; \mathrm{end}
getattribute()
                 305 function luacolor.getattribute()
                 306 return attribute
                 307 end
                 2.9.4 Whatsit insertion
                308 local LIST = 1
                 309 local LIST_LEADERS = 2
                 310 local LIST_DISC = 3
                311 \log a COLOR = 4
                312 local NOCOLOR = 5
                313 local RULE = node.id("rule")
                314 local node_types = {
```

```
315
                 [node.id("hlist")] = LIST,
                 [node.id("vlist")] = LIST,
            316
                 [node.id("rule")] = COLOR,
            317
            318
                 [node.id("glyph")] = COLOR,
                 [node.id("disc")] = LIST_DISC,
            319
                 [node.id("whatsit")] = {
            320
                    [node.subtype("pdf_colorstack")] =
            321
                     function(n)
            322
                       return n.stack == 0 and NOCOLOR or nil
            323
            324
                     end,
            325
                    [node.subtype("special")] = COLOR,
                    [node.subtype("pdf_literal")] = COLOR,
            326
            327
                    [node.subtype("pdf_save")] = COLOR,
            328
                    [node.subtype("pdf_restore")] = COLOR, -- probably not needed
            329 -- TODO (DPC)
                               [node.subtype("pdf_refximage")] = COLOR,
            330 },
                 [node.id("glue")] =
            331
            332
                   function(n)
                     if n.subtype >= 100 then -- leaders
            333
                       if n.leader.id == RULE then
            334
                         return COLOR
            335
                        else
            336
                         return LIST_LEADERS
            337
                        end
            339
                     end
            340
                   end,
            341 }
get_type()
            342 local function get_type(n)
            343 local ret = node_types[n.id]
                if type(ret) == 'table' then
            345
                   ret = ret[n.subtype]
            346
                end
                if type(ret) == 'function' then
            347
                  ret = ret(n)
            348
                 end
            349
            350
                return ret
            351 end
            352 local mode = 2 -- luatex.pdfliteral.direct
            353 local WHATSIT = node.id("whatsit")
            354 local SPECIAL = node.subtype("special")
            355 local PDFLITERAL = node.subtype("pdf_literal")
            356 local DRY_FALSE = false
            357 local DRY_TRUE = true
traverse()
            358 local function traverse(list, color, dry)
                 if not list then
            359
                   return color
            360
            361 end
            362
                 local head
            363
                 if get_type(list) == LIST then
                   head = list.head
            364
            365 elseif get_type(list) == LIST_DISC then
                   head = list.replace
            366
            367 else
                   texio.write_nl("!!! Error: Wrong list type: " .. node.type(list.id))
            368
```

```
370
                                      end
                          371 (debug)texio.write_nl("traverse: " .. node.type(list.id))
                                      for n in node.traverse(head) do
                          373 (debug)texio.write_nl(" node: " .. node.type(n.id))
                                           local t = get_type(n)
                          374
                          375 \langle debug \rangle texio.write\_nl("TYPE "..tostring(t).. " "..tostring(node.type(node.getid(n))).." ".. tostring(t)..." "...tostring(node.type(node.getid(n)))..." "...tostring(t)..." "...tostring(node.type(node.getid(n)))..." "...tostring(t)..." "...tostring(node.type(node.type(node.getid(n)))..." "...tostring(t)..." "...tostring(t)..." "...tostring(node.type(node.type(node.getid(n)))..." "...tostring(t)..." "...tostring(t)...tostring(t)..." "...tostring(t)..." "...tostring(t)...tostring(t)..." "...tostring(t)..." "...tostr
                                           if t == LIST or t == LIST_DISC then
                          376
                                                color = traverse(n, color, dry)
                          377
                                           elseif t == LIST_LEADERS then
                          378
                                                local color_after = traverse(n.leader, color, DRY_TRUE)
                          380
                                                if color == color_after then
                                                     traverse(n.leader, color, DRY_FALSE or dry)
                          381
                          382
                                                else
                                                     traverse(n.leader, '', DRY_FALSE or dry)
                          383
                          The color status is unknown here, because the leader box will or will not be set.
                                                     color = ''
                          384
                                                end
                          385
                                           elseif t == COLOR then
                          386
                                                local v = node.has_attribute(n, attribute)
                          387
                                                if v then
                                                     local newColor = map[v]
                          389
                          390
                                                     if newColor ~= color then
                          391
                                                         color = newColor
                                                         if dry == DRY_FALSE then
                          392
                                                              local newNode
                          393
                                                              if ifpdf then
                          394
                                                                   newNode = node.new(WHATSIT, PDFLITERAL)
                          395
                                                                   newNode.mode = mode
                          396
                                                                   newNode.data = color
                          397
                                                              else
                          398
                                                                  newNode = node.new(WHATSIT, SPECIAL)
                          399
                                                                   newNode.data = prefix .. color
                          400
                           401
                                                              head = node.insert_before(head, n, newNode)
                          402
                          403
                                                          end
                          404
                                                     end
                          405
                                                end
                                           elseif t == NOCOLOR then
                          406
                                                color = ''
                          407
                          408
                                           end
                          409
                                      if get_type(list) == LIST then
                          410
                                           list.head = head
                          411
                          412
                                      else
                                           list.replace = head
                          413
                          414
                                      end
                                     return color
                          415
                          416 \; \mathrm{end}
process()
                          417 function luacolor.process(box)
                          418 local color = ""
                                     local list = tex.getbox(box)
                          420
                                      traverse(list, color, DRY_FALSE)
                          421 end
                          423 if luatexbase.callbacktypes.pre_shipout_filter then
```

369

return color

```
424 luatexbase.add_to_callback("pre_shipout_filter", function(list)
425 traverse(list, "", DRY_FALSE)
426 return true
427 end, "luacolor.process")
428 end
```

For recent versions of luaotfload, we can register a callback to control how coloring glyph is handled for the color feature.

```
429 if luaotfload.set_colorhandler then
430 local set_attribute = node.direct.set_attribute
431 luaotfload.set_colorhandler(function(head, n, color)
432 set_attribute(n, attribute, luacolor.getvalue(color))
433 return head, n
434 end)
435 end
436 ⟨/lua⟩
```

3 Installation

3.1 Download

Package. This package is available on CTAN¹:

CTAN:macros/latex/contrib/luacolor/luacolor.dtx The source file.

CTAN:macros/latex/contrib/luacolor/luacolor.pdf Documentation.

Bundle. All the packages of the bundle 'luacolor' are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

```
CTAN:install/macros/latex/contrib/luacolor.tds.zip
```

TDS refers to the standard "A Directory Structure for TEX Files" (CTAN:pkg/tds). Directories with texmf in their name are usually organized this way.

3.2 Bundle installation

Unpacking. Unpack the luacolor.tds.zip in the TDS tree (also known as texmf tree) of your choice. Example (linux):

```
unzip luacolor.tds.zip -d ~/texmf
```

Script installation. Check the directory TDS:scripts/luacolor/ for scripts that need further installation steps.

3.3 Package installation

Unpacking. The .dtx file is a self-extracting docstrip archive. The files are extracted by running the .dtx through plain TeX:

```
tex luacolor.dtx
```

¹CTAN:pkg/luacolor

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as texmf tree):

```
\label{luacolor.sty} \begin{array}{l} \texttt{luacolor.sty} \to \texttt{tex/latex/luacolor/luacolor.sty} \\ \texttt{luacolor.lua} \to \texttt{scripts/luacolor/luacolor.lua} \\ \texttt{luacolor.pdf} \to \texttt{doc/latex/luacolor/luacolor.pdf} \\ \texttt{luacolor.dtx} \to \texttt{source/latex/luacolor/luacolor.dtx} \end{array}
```

If you have a docstrip.cfg that configures and enables docstrip's TDS installing feature, then some files can already be in the right place, see the documentation of docstrip.

3.4 Refresh file name databases

If your T_EX distribution (T_EX Live, MiKT_EX, ...) relies on file name databases, you must refresh these. For example, T_EX Live users run texhash or mktexlsr.

3.5 Some details for the interested

Unpacking with LATEX. The .dtx chooses its action depending on the format:

plain TEX: Run docstrip and extract the files.

LATEX: Generate the documentation.

If you insist on using LATEX for docstrip (really, docstrip does not need LATEX), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{luacolor.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

Generating the documentation. You can use both the .dtx or the .drv to generate the documentation. The process can be configured by the configuration file ltxdoc.cfg. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with pdfL^AT_EX:

```
pdflatex luacolor.dtx
makeindex -s gind.ist luacolor.idx
pdflatex luacolor.dtx
makeindex -s gind.ist luacolor.idx
pdflatex luacolor.dtx
```

4 History

[2007/12/12 v1.0]

• First public version.

[2009/04/10 v1.1]

• Fixes for changed syntax of \directlua in LuaTeX 0.36.

[2010/03/09 v1.2]

• Adaptation for package luatex 2010/03/09 v0.4.

[2010/12/13 v1.3]

- Support for \pdfxform added.
- Loaded package luatexbase-attr recognized.
- Update for LuaT_FX: 'list' fields renamed to 'head' in v0.65.0.

[2011/03/29 v1.4]

• Avoid whatsit insertion if option monochrome is used (thanks Manuel Pégourié-Gonnard).

[2011/04/22 v1.5]

- Bug fix by Manuel Pégourié-Gonnard: A typo prevented the detection of whatsits and applying color changes for \pdfliteral and \special nodes that might contain typesetting material.
- Bug fix by Manuel Pégourié-Gonnard: Now colors are also applied to leader boxes.
- Unnecessary color settings are removed for leaders boxes, if after the leader box the color has not changed. The costs are a little runtime, leader boxes are processed twice.
- Additional whatsits that are colored: pdf_refximage.
- Workaround for bug with node.insert_before removed for the version after LuaTeX 0.65, because bug was fixed in 0.27. (Thanks Manuel Pégourié-Gonnard.)

[2011/04/23 v1.6]

- Bug fix for nested leader boxes.
- Bug fix for leader boxes that change color, but are not set because of missing place.
- Version check for Lua module added.

[2011/10/22 v1.7]

• Lua functions getattribute and getvalue added to tell other external Lua functions the attribute register number for coloring.

[2011/11/01 v1.8]

• Use of node.subtype instead of magic numbers.

[2016/05/13 v1.9]

- More use of node.subtype instead of magic numbers.
- luatex 85 updates

[2016/05/16 v1.10]

• Documentation updates.

[2018/11/22 v1.11]

- handle issue 43.
- removed pre-0.65 stuff

[2019/07/25 v1.12]

• removed uses of module function, see PR70

[2019/11/29 v1.13]

- Documentation updates.
- Use iftex directly.

[2020-02-22 v1.14]

- Drop use of iftex ltxcmds and infwarerr.
- Assume Itluatex preloaded into format (true since 2015).
- Patch \saveboxresource rather than \pdfxform (keep old name as alias).
- Grab the number via Lua so that a \immediate prefix still works with \saveboxresource/\pdfxform.
- Added handler for the color feature of luaotfload

[2020-02-24 v1.15]

• Grab all possible arguments for \saveboxresource/\pdfxform

[2020-04-04 v1.16]

 \bullet Reset color after ${\tt pdf_colorstack}$ what sits.

[2021-02-17 v1.17]

• Use IFTEX 2ε 's new pre_shipout_filter callback if it's available to allow coloring background and foregrund layer material

5 Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; plain numbers refer to the code lines where the entry is used.

Symbols	\@ne 94, 152
\@ehc 59, 91, 113	\@secondoftwo
\@empty 109	\@undefined 56
\@gobble 119.145	\\ 279, 283

	\- ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
A 100	\LuaCol@scan@boxresource@v . 209, 213
\allocationnumber 126	\LuaCol@scan@number 164, 203, 211, 223
\AtBeginShipout 147	\LuaCol@scan@tobrace 169, 183, 193
\AtBeginShipoutBox 148	\LuaCol@setattribute 124, 129
C	\luacolorProcessBox 2, <u>136</u> , 148, 214
C	\luaescapestring 131
\catcode	M
3, 5, 6, 7, 11, 12, 13, 14, 15, 16,	\MessageBreak 88, 89, 111
17, 20, 21, 23, 24, 25, 26, 30, 32 \csname \cdots \	(Hessagebleak 00, 03, 111
\current@color 97, 131	${f N}$
(current@coror 97, 151	\NeedsTeXFormat 52
D	\newattribute 123
\define@color 70	\number 126, 138
\directlua 56, 73, 77, 79, 105,	,
125, 130, 137, 141, 158, 165, 170	O
\dvidetect() 266	\outputmode 94, 152
_	_
${f E}$	P
\endcsname9	\PackageError 57, 87, 110
\endinput 51	\PackageInfo
\endlinechar 4, 10, 22	\PackageWarningNoLine 64
\expanded . 157, 181, 191, 201, 209, 221	\pdfxform
	\protected 128, 218
G	\ProvidesPackage
\get()	<u>,</u>
\get_type()	${f R}$
\getvalue()	\RequirePackage 55, 146
	\reserved@a 104, 109, 118
\getversion() 236	/reper verse 104, 109, 110
\getversion()	\reset@color
\getversion() 236	\reset@color 68, 98, <u>135</u>
	\reset@color 68, 98, <u>135</u> S
H \hbox	\reset@color 68, 98, <u>135</u> S \saveboxresource 153, 218, 228
H \hbox	\reset@color
H \hbox 99 I \ifcolors@ 62	\reset@color
H \hbox 99 I \ifcolors@ 62 \ifnum 94, 152	\reset@color
H \hbox 99 I \ifcolors@ 62 \ifnum 94, 152 \ifx 56, 85, 109	\reset@color
H \hbox 99 I \ifcolors@ 62 \ifnum 94, 152	\reset@color
H \hbox 99 I \ifcolors@ 62 \ifnum 94, 152 \ifx 56, 85, 109	\reset@color
H \hbox 99 I \ifcolors@ 62 \ifnum 94, 152 \ifx 56, 85, 109 \info() 258	\reset@color
H \hbox 99 I \ifcolors@ 62 \ifnum 94, 152 \ifx 56, 85, 109 \info() 258 L \LuaCol@AtEnd 28, 29, 51, 60, 71, 115, 230	\reset@color
H \hbox 99 I \ifcolors@ 62 \ifnum 94, 152 \ifx 56, 85, 109 \info() 258	\reset@color
H \hbox 99 I \ifcolors@ 62 \ifnum 94, 152 \ifx 56, 85, 109 \info() 258 L \LuaCol@AtEnd 28, 29, 51, 60, 71, 115, 230 \LuaCol@Attribute 123, 129	\reset@color
H \hbox 99 I \ifcolors@ 62 \ifnum 94, 152 \ifx 56, 85, 109 \info() 258 L \LuaCol@AtEnd 28, 29, 51, 60, 71, 115, 230 \LuaCol@Attribute 123, 129 \LuaCol@if@keyword	S \saveboxresource
H \hbox 99 I \ifcolors@ 62 \ifnum 94, 152 \ifx 56, 85, 109 \info() 258 L \LuaCol@AtEnd 28, 29, 51, 60, 71, 115, 230 \LuaCol@Attribute 123, 129 \LuaCol@if@keyword 156, 180, 190, 200, 219	\reset@color
H \hbox 99 I \ifcolors@ 62 \ifnum 94, 152 \ifx 56, 85, 109 \info() 258 L \LuaCol@AtEnd 28, 29, 51, 60, 71, 115, 230 \LuaCol@Attribute 123, 129 \LuaCol@i@keyword 156, 180, 190, 200, 219 \LuaCol@i@i 155	S \saveboxresource
H \hbox 99 I \ifcolors@ 62 \ifnum 94, 152 \ifx 56, 85, 109 \info() 258 L \LuaCol@AtEnd 28, 29, 51, 60, 71, 115, 230 \LuaCol@Attribute 123, 129 \LuaCol@if@keyword 156, 180, 190, 200, 219 \LuaCol@ii@i 155 \LuaCol@iii@i@ii 155 \LuaCol@org@pdfxform 153, 215 \LuaCol@scan@boxresource@i	\reset@color
H \hbox 99 I \ifcolors@ 62 \ifnum 94, 152 \ifx 56, 85, 109 \info() 258 L \LuaCol@AtEnd 28, 29, 51, 60, 71, 115, 230 \LuaCol@Attribute 123, 129 \LuaCol@if@keyword 156, 180, 190, 200, 219 \LuaCol@ii@i 155 \LuaCol@ii@ii@ii 155 \LuaCol@ii@ii@ii 154, 157 \LuaCol@org@pdfxform 153, 215 \LuaCol@scan@boxresource@i 179, 221, 225	\reset@color
H \hbox 99 I \ifcolors@ 62 \ifnum 94, 152 \ifx 56, 85, 109 \info() 258 L \LuaCol@AtEnd 28, 29, 51, 60, 71, 115, 230 \LuaCol@Attribute 123, 129 \LuaCol@if@keyword 156, 180, 190, 200, 219 \LuaCol@ii@i 155 \LuaCol@ii@i@i 155 \LuaCol@ii@i@ii 154, 157 \LuaCol@org@pdfxform 153, 215 \LuaCol@scan@boxresource@i 179, 221, 225 \LuaCol@scan@boxresource@iI 181, 188	\reset@color
H \hbox 99 I \ifcolors@ 62 \ifnum 94, 152 \ifx 56, 85, 109 \info() 258 L \LuaCol@AtEnd 28, 29, 51, 60, 71, 115, 230 \LuaCol@Attribute 123, 129 \LuaCol@if@keyword 156, 180, 190, 200, 219 \LuaCol@ii@i 155 \LuaCol@ii@ii@ii 154, 157 \LuaCol@ii@ii@iii 154, 157 \LuaCol@org@pdfxform 153, 215 \LuaCol@scan@boxresource@i 179, 221, 225 \LuaCol@scan@boxresource@ii 181, 188 \LuaCol@scan@boxresource@ii	S \saveboxresource 153, 218, 228 \set@color 67, 101, 128, 151 \set@page@color 69 \setattribute 124 \setattribute() 301 \setbox 99 \special 112, 118 T \the 10, 11, 12, 13, 14, 15, 16, 17, 30 \TMP@EnsureCode 27, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50 \traverse() 358 U \unexpanded 157, 181, 191, 201, 209, 221
H \hbox 99 I \ifcolors@ 62 \ifnum 94, 152 \ifx 56, 85, 109 \info() 258 L \LuaCol@AtEnd 28, 29, 51, 60, 71, 115, 230 \LuaCol@Attribute 123, 129 \LuaCol@if@keyword 156, 180, 190, 200, 219 \LuaCol@ii@i 155 \LuaCol@ii@i 155 \LuaCol@ii@iii 154, 157 \LuaCol@org@pdfxform 153, 215 \LuaCol@scan@boxresource@i 179, 221, 225 \LuaCol@scan@boxresource@ii 181, 188 \LuaCol@scan@boxresource@ii 185, 188, 189	S \saveboxresource
H \hbox 99 I \ifcolors@ 62 \ifnum 94, 152 \ifx 56, 85, 109 \info() 258 L \LuaCol@AtEnd 28, 29, 51, 60, 71, 115, 230 \LuaCol@Attribute 123, 129 \LuaCol@if@keyword 156, 180, 190, 200, 219 \LuaCol@ii@i 155 \LuaCol@ii@ii 155 \LuaCol@ii@iii 154, 157 \LuaCol@org@pdfxform 153, 215 \LuaCol@org@pdfxform 153, 215 \LuaCol@scan@boxresource@i 179, 221, 225 \LuaCol@scan@boxresource@ii 185, 188, 189 \LuaCol@scan@boxresource@ii 185, 188, 189 \LuaCol@scan@boxresource@ii 185, 188, 189 \LuaCol@scan@boxresource@ii 185, 188, 189	S \saveboxresource
H \hbox 99 I \ifcolors@ 62 \ifnum 94, 152 \ifx 56, 85, 109 \info() 258 L \LuaCol@AtEnd 28, 29, 51, 60, 71, 115, 230 \LuaCol@Attribute 123, 129 \LuaCol@if@keyword 156, 180, 190, 200, 219 \LuaCol@ii@i 155 \LuaCol@ii@i 155 \LuaCol@ii@ii 154, 157 \LuaCol@org@pdfxform 153, 215 \LuaCol@org@pdfxform 153, 215 \LuaCol@scan@boxresource@i 179, 221, 225 \LuaCol@scan@boxresource@ii 185, 188, 189 \LuaCol@scan@boxresource@iii 185, 188, 189 \LuaCol@scan@boxresource@iii 185, 188, 189 \LuaCol@scan@boxresource@iii 181, 198 \LuaCol@scan@boxresource@iii	S \saveboxresource
H \hbox 99 I \ifcolors@ 62 \ifnum 94, 152 \ifx 56, 85, 109 \info() 258 L \LuaCol@AtEnd 28, 29, 51, 60, 71, 115, 230 \LuaCol@Attribute 123, 129 \LuaCol@if@keyword 156, 180, 190, 200, 219 \LuaCol@ii@i 155 \LuaCol@ii@i 155 \LuaCol@ii@ii 154, 157 \LuaCol@org@pdfxform 153, 215 \LuaCol@org@pdfxform 153, 215 \LuaCol@scan@boxresource@i 179, 221, 225 \LuaCol@scan@boxresource@ii 185, 188, 189 \LuaCol@scan@boxresource@ii 185, 188, 189 \LuaCol@scan@boxresource@ii 191, 198 \LuaCol@scan@boxresource@ii 195, 198, 199	S Saveboxresource
H \hbox 99 I \ifcolors@ 62 \ifnum 94, 152 \ifx 56, 85, 109 \info() 258 L \LuaCol@AtEnd 28, 29, 51, 60, 71, 115, 230 \LuaCol@Attribute 123, 129 \LuaCol@if@keyword 156, 180, 190, 200, 219 \LuaCol@ii@i 155 \LuaCol@ii@i 155 \LuaCol@ii@ii 154, 157 \LuaCol@org@pdfxform 153, 215 \LuaCol@org@pdfxform 153, 215 \LuaCol@scan@boxresource@i 179, 221, 225 \LuaCol@scan@boxresource@ii 185, 188, 189 \LuaCol@scan@boxresource@iii 185, 188, 189 \LuaCol@scan@boxresource@iii 185, 188, 189 \LuaCol@scan@boxresource@iii 181, 198 \LuaCol@scan@boxresource@iii	S \saveboxresource