

The hycolor package

Heiko Oberdiek*

2020-01-27 v1.10

Abstract

Package hycolor implements the color option stuff that is used by packages hyperref and bookmark. It is not intended as package for the user.

Contents

1	Documentation	2
1.1	Summary	2
2	Implementation	3
2.1	Normalization	4
2.1.1	Sanitize value of color option	4
2.1.2	Normalize result	5
2.2	Main algorithm for color options	7
2.3	Package bookmark	7
2.4	Utils	9
2.5	Package hyperref	10
2.5.1	Options Hyp.*color	10
2.5.2	Generic algorithm	12
2.5.3	Field options	14
2.5.4	Detection for naked RGB values	14
2.5.5	Options *bordercolor	16
2.6	Package attachfile2	17
2.7	Patch for package xcolor	19
2.7.1	Fix fragile \@frameb@x	22
3	Installation	23
3.1	Download	23
3.2	Package installation	23
3.3	Refresh file name databases	23
3.4	Some details for the interested	23
4	History	24
	[2007/04/09 v1.0]	24
	[2007/04/11 v1.1]	24
	[2008/07/29 v1.2]	24
	[2008/08/01 v1.3]	24
	[2008/09/08 v1.4]	24
	[2009/10/02 v1.5]	24
	[2009/12/12 v1.6]	24

*Please report any issues at <https://github.com/ho-tex/hycolor/issues>

[2011/01/30 v1.7]	24
[2016/05/16 v1.8]	24
[2019/12/15 v1.9]	25
[2020-01-27 v1.10]	25

5 Index **25**

1 Documentation

The package `hycolor` implements color options for packages `hyperref` and `bookmark`.

Package `xcolor` provides macros for extracting color values and converting color data to other color models. If this package is loaded, the full range of color specifications of packages `color` and `xcolor` are supported including the optional argument for the color model.

```
\hypersetup{linkbordercolor=red}% needs xcolor
\hypersetup{linkbordercolor=[named]{red}}% needs xcolor
\hypersetup{linkbordercolor=[rgb]{1,0,0}}
```

Without package `xcolor` some of the options only support some models, if they are given directly, e.g.:

```
\bookmarksetup{color=[rgb]{1,0,0}}
```

Because of compatibility some options of `hyperref` also support space separated RGB values:

```
\hypersetup{linkbordercolor=1 0 0}% is the same as
\hypersetup{linkbordercolor=[rgb]{1,0,0}}
```

Coloring is optional, it can be turned off by using an empty value:

```
\hypersetup{linkbordercolor={}}
```

The PDF specification knows some kind of an empty color setting without values. This applies to form field colors. The new A virtual color model `empty` is introduced for this purpose, e.g.

```
\TextField[backgroundcolor={ [empty] {}}, ...]{...}% or
\TextField[{backgroundcolor=[empty]{}, ...}]{...}
```

PDF specification 1.7 also allows this for border link colors. But this isn't currently supported by this package, because the tested viewers (AR7/Linux, xpdf 3.00, ghostscript 8.54) don't support this yet. In contrary ghostscript generates an error message.

1.1 Summary

Color option	Models without xcolor	RGB color	Model empty
<code>BKM.color</code>	gray, rgb	no	no
<code>Hyp.*color</code>	all	no	no
<code>Hyp.*bordercolor</code>	gray, rgb	yes	no
<code>Field.*color</code>	gray, rgb, cmyk	yes	yes
<code>AtFi.color</code>	gray, rgb	yes	no

“RGB color” means that the color value can be given as space separated RGB numbers (real numbers in the range from 0 to 1). Explanation of the color option prefixes:

Prefix	Explanation
BKM	Package bookmark
Hyp	Package hyperref: package options or <code>\hypersetup</code>
Field	Package hyperref: Form field options
AtFi	Package attachfile2: option color

2 Implementation

```

1 (*package)
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{hycolor}%
4 [2020-01-27 v1.10 Color options for hyperref/bookmark (HO)]%

```

Should not be needed after xcolor updates, Avoid loading xcolor-patch but fix Gray color model.

```

5 % \RequirePackage{xcolor-patch}[2019/12/15]
6 \AtBeginDocument{%
7 \def\@tempa#1,{%
8   \XC@ifxcase\tm{%
9     \XC@mod@rgb{%
10      \XC@calcN{#1}\@tmp
11      \edef\@tmp{\@tmp,\@tmp,\@tmp}%
12    }%
13    \XC@mod@cmy{%
14      \XC@calcC{#1}\@tmp
15      \edef\@tmp{\@tmp,\@tmp,\@tmp}%
16    }%
17    \XC@mod@cmyk{%
18      \XC@calcC{#1}\@tmp
19      \edef\@tmp{0,0,0,\@tmp}%
20    }%
21    \XC@mod@RGB{%
22      \edef\@sc1{\rangeRGB}%
23      \XC@calcM{#1}\@tmp
24      \edef\@tmp{\@tmp,\@tmp,\@tmp}%
25    }%
26    \XC@mod@HTML{%
27      \edef\@sc1{\@cclv}%
28      \XC@calcM{#1}\@tmp
29      \XC@calcH\@tmp\@tmp
30      \edef\@tmp{\@tmp\@tmp\@tmp}%
31    }%
32    \XC@mod@HSB{%
33      \edef\@sc1{\rangeHSB}%
34      \XC@calcM{#1}\@tmp
35      \edef\@tmp{0,0,\@tmp}%
36    }%
37    \XC@mod@Gray{%
38      \edef\@sc1{\rangeGray}%
39      \XC@calcM{#1}\@tmp
40    }%
41  }%
42  {%
43    \XC@calcN{#1}\@tmp
44    \edef\@tmp{0,0,\@tmp}%
45  }%
46 }%
47 \ifx\XC@cnv@gray\@tempa

```

```

48 \def\XC@cnv@gray#1,{%
49 \XC@ifxcase\tm{%
50 \XC@mod@rgb{%
51 \XC@calcN{#1}\@tmp
52 \edef\@tmp{\@tmp,\@tmp,\@tmp}%
53 }%
54 \XC@mod@gray{%
55 \XC@mod@cmy{%
56 \XC@calcC{#1}\@tmp
57 \edef\@tmp{\@tmp,\@tmp,\@tmp}%
58 }%
59 \XC@mod@cmyk{%
60 \XC@calcC{#1}\@tmp
61 \edef\@tmp{0,0,0,\@tmp}%
62 }%
63 \XC@mod@RGB{%
64 \edef\@scl{\rangeRGB}%
65 \XC@calcM{#1}\@tmp
66 \edef\@tmp{\@tmp,\@tmp,\@tmp}%
67 }%
68 \XC@mod@HTML{%
69 \edef\@scl{\@cclv}%
70 \XC@calcM{#1}\@tmp
71 \XC@calcH\@tmp\@tmp
72 \edef\@tmp{\@tmp\@tmp\@tmp}%
73 }%
74 \XC@mod@HSB{%
75 \edef\@scl{\rangeHSB}%
76 \XC@calcM{#1}\@tmp
77 \edef\@tmp{0,0,\@tmp}%
78 }%
79 \XC@mod@Gray{%
80 \edef\@scl{\rangeGray}%
81 \XC@calcM{#1}\@tmp
82 }%
83 }%
84 {%
85 \XC@calcN{#1}\@tmp
86 \edef\@tmp{0,0,\@tmp}%
87 }%
88 }%
89 \fi
90 \let\@tempa\relax
91 }

```

2.1 Normalization

2.1.1 Sanitize value of color option

Procedure DefSanitized(cmd, value)

Param: *cmd* (macro)

Param: *value* (value of color option)

Result: *value* is expanded, sanitized, and stored in macro *cmd*.

Initialize active characters;

cmd := Expand *value*;

Sanitize *cmd*;

Sanitization means that the string does not contain any macros or special

tokens. It consists of characters with catcode 12 (other). The only exception is the space with catcode 10 (space).

`\HyColor@DefSanitized`

```

92 \begingroup
93 \catcode'\!=13 %
94 \catcode'\:=13 %
95 \catcode'\-=13 %
96 \catcode'\+=13 %
97 \catcode'\;=13 %
98 \catcode'\ "=13 %
99 \catcode'\>=13 %
100 \edef\x{%
101 \def\noexpand!\string!}%
102 \def\noexpand:\string:}%
103 \def\noexpand-\string-}%
104 \def\noexpand+\string+}%
105 \def\noexpand;\string;}%
106 \def\noexpand"\string"%}
107 \def\noexpand>\string>}%
108 }%
109 \def\y#1{\endgroup
110 \def\HyColor@DefSanitized##1##2{%
111 \begingroup
112 \csname @safe@activetrue\endcsname
113 #1%
114 \edef\x{\endgroup
115 \def\noexpand##1{##2}%
116 }%
117 \x
118 \@onelevel@sanitize##1%
119 }%
120 }%
121 \expandafter\y\expandafter{x}

```

2.1.2 Normalize result

Procedure `NormalizeNum(value, cmd)`

Param: *value* (Sanitized explicit number)

Param: *cmd* (Macro that stores result)

Result: *cmd* contains normalized number

if *value* pt < 0pt **then**

 | *cmd* ← 0;

else if *number before dot of value* < 1 **then**

 | *cmd* ← number after dot of *value*;

 | *cmd* ← strip trailing zeros from *cmd*;

if *dot remains only* **then**

 | *cmd* ← 0;

end

else

 | *cmd* ← 1;

end

The number is limited to the range between 0.0 and 1.0 and formatted as short PDF number without leading or trailing zeros. The precision of the number isn't changed.

\HyColor@NormalizeNum

```
122 \def\HyColor@NormalizeNum#1#2{%
123   \ifdim#1pt<\z@
124     \def#2{0}%
125   \else
126     \edef#2{\zap@space#1 \@empty}%
127     \expandafter\HyColor@CheckDot#2..\@nil#2%
128   \fi
129 }
130 \def\HyColor@CheckDot#1.#2.#3\@nil#4{%
131   \ifnum0#1<\@ne
132     \ifx\#2\%
133       \def#4{0}%
134     \else
135       \edef#4{\HyColor@ReverseString#2\@nil{}}%
136       \edef#4{\expandafter\HyColor@StripLeadingZeros#4\@empty}%
137       \ifx#4\@empty
138         \def#4{0}%
139       \else
140         \edef#4{.\expandafter\HyColor@ReverseString#4\@nil{}}%
141       \fi
142     \fi
143   \else
144     \def#4{1}%
145   \fi
146 }
147 \def\HyColor@ReverseString#1#2\@nil#3{%
148   \ifx\#2\%
149     #1#3%
150   \else
151     \@ReturnAfterFi{%
152       \HyColor@ReverseString#2\@nil{#1#3}%
153     }%
154   \fi
155 }
156 \long\def\@ReturnAfterFi#1\fi{\fi#1}
157 \def\HyColor@StripLeadingZeros#1{%
158   \ifx#10%
159     \expandafter\HyColor@StripLeadingZeros
160   \else
161     #1%
162   \fi
163 }
```

\HyColor@NormalizeCommaRGB

```
164 \def\HyColor@NormalizeCommaRGB#1,#2,#3\@nil#4{%
165   \HyColor@NormalizeNum{#1}\HyColor@temp
166   \let#4\HyColor@temp
167   \HyColor@NormalizeNum{#2}\HyColor@temp
168   \edef#4{#4 \HyColor@temp}%
169   \HyColor@NormalizeNum{#3}\HyColor@temp
170   \edef#4{#4 \HyColor@temp}%
171 }
```

\HyColor@NormalizeCommaCMYK

```
172 \def\HyColor@NormalizeCommaCMYK#1,#2,#3,#4\@nil#5{%
173   \HyColor@NormalizeNum{#1}\HyColor@temp
174   \let#5\HyColor@temp
```

```

175 \HyColor@NormalizeNum{#2}\HyColor@temp
176 \edef#5{#5 \HyColor@temp}%
177 \HyColor@NormalizeNum{#3}\HyColor@temp
178 \edef#5{#5 \HyColor@temp}%
179 \HyColor@NormalizeNum{#4}\HyColor@temp
180 \edef#5{#5 \HyColor@temp}%
181 }

```

2.2 Main algorithm for color options

Procedure MainColorOptionAlgorithm(key, value, cmd)

Param: *key* (name of color option)

Param: *value* (value of color option)

Param: *cmd* (macro that stores result)

Result: Macro *cmd* contains the calculated color specification string or has the meaning of `\relax` if the color must not set

DefSanitized(*temp*, *value*);

Call option specific algorithm(*key*, *temp*, *cmd*);

2.3 Package bookmark

Since v0.8 2007/03/27 package `bookmark` only provides one color option `color`. Because option `rgbcolor` can easily given as color specification in model `rgb`:

$$\text{rgbcolor}=\langle r \rangle \langle g \rangle \langle b \rangle \equiv \text{color}=[\text{rgb}]\{\langle r \rangle, \langle g \rangle, \langle b \rangle\}$$

Package `bookmark` stores the result in macro `\BKM@color`. The empty string is

interpreted as *no color*.

Procedure BookmarkColor(value, cmd, package, option)

Param: *value* (value of option color)

Param: *cmd* (macro for result)

Param: *package* (package name for error message)

Param: *option* (option name for error message)

```
switch value do
  case empty do
    | cmd ← no color;
  end
  case with model do
    | if with xcolor then
      | cmd ← ConvertToRGB(model, values);
    else
      | if model = rgb then
        | cmd ← values as normalized values;
      else if model = gray then
        | cmd ← values as normalized tripled values;
      else
        | error;
      end
    end
  end
end
otherwise do
  | if with xcolor then
    | (model, values ← get model and values;
    | cmd ← ConvertToRGB(model, values);
  else
    | error;
  end
end
end
```

```
182 \def\HyColor@BookmarkColor#1#2#3#4{%
183   \HyColor@IfModel{#1}{%
184     \HyColor@IfXcolor{%
185       \convertcolorspec\HyColor@model\HyColor@values
186         \HyColor@model@rgb#2%
187       \expandafter\HyColor@NormalizeCommaRGB#2\@nil#2%
188     }{%
189       \ifx\HyColor@model\HyColor@model@rgb
190         \expandafter\HyColor@NormalizeCommaRGB\HyColor@values\@nil#2%
191       \else
192         \ifx\HyColor@model\HyColor@model@gray
193           \expandafter\HyColor@NormalizeNum
194           \expandafter{\HyColor@values}#2%
195         \edef#2{#2 #2 #2}%
196       \else
197         \let#2\@empty
198         \HyColor@ErrorModelNoXcolor{#3}{#4}%
199       \fi
200     \fi
201   }%
202 }{%
```

```

203 \let#2\HyColor@values
204 \ifx#2\@empty
205 \else
206 \HyColor@IfXcolor{%
207 \extractcolorspec{#1}#2%
208 \expandafter\convertcolorspec#2\HyColor@model@rgb#2%
209 \expandafter\HyColor@NormalizeCommaRGB#2\@nil#2%
210 }{%
211 \let#2\@empty
212 \HyColor@ErrorSpecNoXcolor{#3}{#4}%
213 }%
214 \fi
215 }%
216 }

217 \def\HyColor@ErrorModelNoXcolor#1#2{%
218 \PackageError{#1}{%
219 Color model '\HyColor@model' is not supported\MessageBreak
220 without package 'xcolor' in\MessageBreak
221 '#2=[\HyColor@model]{\HyColor@values}'%
222 }\@ehc
223 }

224 \def\HyColor@ErrorSpecNoXcolor#1#2{%
225 \PackageError{#1}{%
226 This color specification is not supported\MessageBreak
227 without package 'xcolor' in\MessageBreak
228 '#2=\HyColor@values'%
229 }\@ehc
230 }

231 \def\HyColor@IfModel#1{%
232 \@ifnextchar[{%
233 \HyColor@WithModel
234 }{%
235 \HyColor@WithoutModel
236 }%
237 #1\@nil
238 }

239 \def\HyColor@WithModel[#1]#2\@nil{%
240 \HyColor@DefSanitized\HyColor@model{#1}%
241 \HyColor@DefSanitized\HyColor@values{#2}%
242 \@firstoftwo
243 }

244 \def\HyColor@WithoutModel#1\@nil{%
245 \let\HyColor@model\relax
246 \HyColor@DefSanitized\HyColor@values{#1}%
247 \@secondoftwo
248 }

```

2.4 Utils

\@ReturnAfterFi

```
249 \long\def\@ReturnAfterFi#1\fi{\fi#1}
```

\HyColor@IfXcolor

```

250 \def\HyColor@IfXcolor{%
251 \begingroup\expandafter\expandafter\endgroup
252 \expandafter\ifx\csname convertcolorspec\endcsname\relax

```

```

253   \expandafter\@secondoftwo
254   \else
255   \expandafter\@firstoftwo
256   \fi
257 }

258 \def\HyColor@model@empty{empty}
259 \@onelevel@sanitize\HyColor@model@empty
260 \def\HyColor@model@gray{gray}
261 \@onelevel@sanitize\HyColor@model@gray
262 \def\HyColor@model@rgb{rgb}
263 \@onelevel@sanitize\HyColor@model@rgb
264 \def\HyColor@model@cmyk{cmyk}
265 \@onelevel@sanitize\HyColor@model@cmyk
266 \def\HyColor@model@Gray{Gray}
267 \@onelevel@sanitize\HyColor@model@Gray

```

2.5 Package hyperref

2.5.1 Options Hyp.*color

```

268 \def\HyColor@UseColor#1{%
269   \ifx\relax#1\@empty
270   \else
271   \ifx\@empty#1\@empty
272   \else
273     \expandafter\expandafter\expandafter\HyColor@@UseColor#1\@nil
274   \fi
275   \fi
276 }
277 \def\HyColor@@UseColor{%
278   \@ifnextchar[\HyColor@@@UseColor\HyColor@@@UseColor
279 }
280 \def\HyColor@@@UseColor[#1]#2\@nil{%
281   \color[#{#1}]{#2}%
282 }
283 \def\HyColor@@@UseColor#1\@nil{%
284   \color{#1}%
285 }

```

Procedure HyperrefColor(value, cmd)

Param: *value* (value of the option)

Param: *cmd* (macro for result)

```

switch value do
| case empty do
|   cmd ← no color;
| end
| case with model do
|   Call \color with model;
| end
| case without model do
|   Call \color without model;
| end
end
end

```

```

286 \def\HyColor@HyperrefColor#1#2{%
287   \HyColor@IfModel{#1}{%
288     \edef#2[{\HyColor@model}]{\HyColor@values}}%

```

```
289 }{%
290   \let#2\HyColor@values
291   \ifx#2\@empty
292     \let#2\relax
293   \fi
294 }%
295 }
```

2.5.2 Generic algorithm

Procedure Algorithm X0134(*value*, *cmd*, *package*, *option*)

Param: *value* (value of the option)

Param: *cmd* (macro for result)

Param: *package* (package name for error message)

Param: *option* (option name for error message)

```
switch value do
  case empty do
    | cmd ← no color;
  end
  case with model do
    switch model do
      case empty do
        | cmd ← "";
      end
      case gray, rgb, cmyk do
        | cmd ← output();
      end
      case Gray do
        if with xcolor then
          | (model, values) ← convert to gray;
        else
          | error(package, option, "Missing xcolor"), cmd ← no color;
        end
      end
      else
        if with xcolor then
          | (model, values) ← convert to rgb;
          | cmd ← output();
        else
          | error(package, option, "Missing xcolor"), cmd ← no color;
        end
      end
    end
  end
  case rgb values do
    | (model, values) ← ("rgb", (r,g,b));
    | cmd ← output();
  end
  case without model do
    if with xcolor then
      | (model, values) ← get model and values(value);
      switch model do
        case gray, rgb, cmyk do
          | cmd ← output();
        end
        case Gray do
          | (model, values) ← convert to gray;
          | cmd ← output();
        end
        else
          | (model, values) ← convert to rgb;
          | cmd ← output();
        end
      end
    else
      | error(package, option, "Missing xcolor"), cmd ← no color;
    end
  end
end
```

\HyColor@XZeroOneThreeFour

296 \def\HyColor@XZeroOneThreeFour#1#2#3#4{%

```

297 \HyColor@IfModel{#1}{%
298   \ifx\HyColor@model\HyColor@model@empty
299     \let#2\@empty
300   \else\ifx\HyColor@model\HyColor@model@gray
301     \expandafter\HyColor@NormalizeNum
302     \expandafter{\HyColor@values}#2%
303   \else\ifx\HyColor@model\HyColor@model@rgb
304     \expandafter\HyColor@NormalizeCommaRGB\HyColor@values\@nil#2%
305   \else\ifx\HyColor@model\HyColor@model@cmyk
306     \expandafter\HyColor@NormalizeCommaCMYK\HyColor@values\@nil#2%
307   \else\ifx\HyColor@model\HyColor@model@Gray
308     \HyColor@IfXcolor{%
309       \convertcolorspec\HyColor@model\HyColor@values
310         \HyColor@model@gray#2%
311       \expandafter\HyColor@NormalizeNum\expandafter{#2}#2%
312       \let\HyColor@model\HyColor@model@gray
313     }{%
314       \let#2\relax
315       \HyColor@ErrorModelNoXcolor{#3}{#4}%
316     }%
317   \else
318     \HyColor@IfXcolor{%
319       \convertcolorspec\HyColor@model\HyColor@values
320         \HyColor@model@rgb#2%
321       \expandafter\HyColor@NormalizeCommaRGB#2\@nil#2%
322       \let\HyColor@model\HyColor@model@rgb
323     }{%
324       \let#2\relax
325       \HyColor@ErrorModelNoXcolor{#3}{#4}%
326     }%
327   \fi\fi\fi\fi\fi
328 }{%
329   \let#2\HyColor@values
330   \ifx#2\@empty
331     \let#2\relax
332   \else
333     \expandafter\HyColor@IfRGB\expandafter{\HyColor@values}{%
334       \expandafter\HyColor@NormalizeCommaRGB\HyColor@values\@nil#2%
335     }{%
336       \HyColor@IfXcolor{%
337         \expandafter\extractcolorspec\expandafter{\HyColor@values}#2%
338         \edef\HyColor@model{\expandafter\@firstoftwo#2}%
339         \edef\HyColor@values{\expandafter\@secondoftwo#2}%
340         \ifx\HyColor@model\HyColor@model@gray
341           \expandafter\HyColor@NormalizeNum\expandafter
342             {\HyColor@values}#2%
343         \else\ifx\HyColor@model\HyColor@model@rgb
344           \expandafter\HyColor@NormalizeCommaRGB
345             \HyColor@values\@nil#2%
346         \else\ifx\HyColor@model\HyColor@model@cmyk
347           \expandafter\HyColor@NormalizeCommaCMYK
348             \HyColor@values\@nil#2%
349         \else\ifx\HyColor@model\HyColor@model@Gray
350           \convertcolorspec\HyColor@model\HyColor@values
351             \HyColor@model@gray#2%
352           \expandafter\HyColor@NormalizeNum\expandafter
353             {\HyColor@values}#2%
354           \let\HyColor@model\HyColor@model@gray

```

```

355         \else
356             \convertcolorspec\HyColor@model\HyColor@values
357             \HyColor@model@rgb#2%
358             \expandafter\HyColor@NormalizeCommaRGB#2\@nil#2%
359             \let\HyColor@model\HyColor@model@rgb
360             \fi\fi\fi\fi
361         }{%
362             \let#2\relax
363             \HyColor@ErrorSpecNoXcolor{#3}{#4}%
364         }%
365     }%
366 \fi
367 }%
368 }

```

2.5.3 Field options

\HyColor@FieldBColor

```

369 \let\HyColor@FieldBColor\HyColor@XZeroOneThreeFour

```

\HyColor@FieldColor

```

370 \def\HyColor@FieldColor#1#2#3#4{%
371     \let\HyColor@model\@empty
372     \HyColor@XZeroOneThreeFour{#1}{#2}{#3}{#4}%
373     \ifx#2\relax
374         \let#2\@empty
375     \else
376         \ifx#2\@empty
377         \else
378             \ifx\HyColor@model\HyColor@model@gray
379                 \edef#2{#2 g}%
380             \else\ifx\HyColor@model\HyColor@model@rgb
381                 \edef#2{#2 rg}%
382             \else\ifx\HyColor@model\HyColor@model@cmyk
383                 \edef#2{#2 k}%
384             \else
385                 \PackageError{#3}{Internal error: unsupported color model}\@ehc
386             \fi\fi\fi
387         \fi
388     \fi
389 }

```

2.5.4 Detection for naked RGB values

\HyColor@IfRGB

```

390 \newif\ifHyColor@result
391 \begingroup\expandafter\expandafter\expandafter\endgroup
392 \expandafter\ifx\csname pdfmatch\endcsname\relax
393     \expandafter\@firstoftwo
394 \else
395     \expandafter\@secondoftwo
396 \fi
397 {%
398     \begingroup
399     \def\x#1{\endgroup
400         \def\HyColor@IfRGB##1{%
401             \HyColor@@IfRGB##1#1#1\@nil
402         }%

```

```

403 }%
404 \x{ }%
405 \edef\HyColor@TwoSpaces{\space\space}%
406 \def\HyColor@@IfRGB#1 #2 #3 #4\@nil{%
407   \HyColor@resulttrue
408   \def\HyColor@temp{#4}%
409   \ifx\HyColor@temp\HyColor@TwoSpaces
410     \HyColor@CheckNum{#1}%
411     \ifHyColor@result
412       \HyColor@CheckNum{#2}%
413       \ifHyColor@result
414         \HyColor@CheckNum{#3}%
415       \fi
416     \fi
417   \else
418     \HyColor@resultfalse
419   \fi
420   \ifHyColor@result
421     \let\HyColor@model\HyColor@model@rgb
422     \edef\HyColor@values{#1,#2,#3}%
423     \expandafter\@firstoftwo
424   \else
425     \expandafter\@secondoftwo
426   \fi
427 }%
428 \def\HyColor@zero{0}%
429 \def\HyColor@one{1}%
430 \def\HyColor@dot{.}%
431 \def\HyColor@CheckNum#1{%
432   \def\HyColor@temp{#1}%
433   \ifx\HyColor@temp\@empty
434     \HyColor@resultfalse
435   \else
436     \edef\HyColor@temp{\@car#1\@nil}%
437     \ifx\HyColor@temp\HyColor@zero
438     \else
439       \ifx\HyColor@temp\HyColor@one
440       \else
441         \ifx\HyColor@temp\HyColor@dot
442         \else
443           \HyColor@resultfalse
444         \fi
445       \fi
446     \fi
447   \fi
448 }%
449 }{%
450 \def\HyColor@MatchNum{%
451   (0*1\string\.0*|0*1|0+\string\.[0-9]*|\string\.[0-9]+)%
452 }%
453 \def\HyColor@IfRGB#1{%
454   \ifnum\pdfmatch{~\HyColor@MatchNum\space\HyColor@MatchNum
455     \space\HyColor@MatchNum$}{#1}>\z@
456   \let\HyColor@model\HyColor@model@rgb
457   \edef\HyColor@values{%
458     \expandafter\strip@prefix\pdfmatch1,%
459     \expandafter\strip@prefix\pdfmatch2,%
460     \expandafter\strip@prefix\pdfmatch3%

```

```

461     }%
462     \HyColor@resulttrue
463     \expandafter\@firstoftwo
464     \else
465     \HyColor@resultfalse
466     \expandafter\@secondoftwo
467     \fi
468 }%
469 }

```

2.5.5 Options ***bordercolor**

Procedure `HyperrefBorderColor`(*value*, *cmd*, *package*, *option*)

Param: *value* (value of the option)

Param: *cmd* (macro for result)

Param: *package*, *option* (package and option for error message)

```

switch value do
  case empty do
    | cmd ← no color;
  end
  case with model do
    | if with xcolor then
      | (model, values) ← convert to rgb;
      | cmd ← output values;
    else
      | switch model do
        | case rgb, gray do
          | cmd ← output values;
        end
        | else
          | error(package, option, "Missing xcolor");
          | cmd ← no color;
        end
      end
    end
  end
  case rgb values do
    | cmd ← output values;
  end
  case without model do
    | if with xcolor then
      | (model, values) ← convert to rgb;
      | cmd ← output values;
    else
      | error(package, option, "Missing xcolor"); cmd ← no color;
    end
  end
end
end

```

`\HyColor@HyperrefBorderColor`

```

470 \def\HyColor@HyperrefBorderColor#1#2#3#4{%
471   \HyColor@IfModel{#1}{%
472     \HyColor@IfXcolor{%

```

```

473     \convertcolorspec\HyColor@model\HyColor@values
474         \HyColor@model@rgb#2%
475     \expandafter\HyColor@NormalizeCommaRGB#2\@nil#2%
476 }{%
477     \ifx\HyColor@model\HyColor@model@rgb
478     \expandafter\HyColor@NormalizeCommaRGB\HyColor@values\@nil#2%
479     \else
480     \ifx\HyColor@model\HyColor@model@gray
481     \expandafter\HyColor@NormalizeNum
482     \expandafter{\HyColor@values}#2%
483     \edef#2{#2 #2 #2}%
484     \else
485     \let#2\relax
486     \HyColor@ErrorModelNoXcolor{#3}{#4}%
487     \fi
488 \fi
489 }%
490 }{%
491     \let#2\HyColor@values
492     \ifx#2\@empty
493     \let#2\relax
494     \else
495     \expandafter\HyColor@IfRGB\expandafter{\HyColor@values}{%
496     \expandafter\HyColor@NormalizeCommaRGB\HyColor@values\@nil#2%
497     }{%
498     \HyColor@IfXcolor{%
499     \extractcolorspec{#1}#2%
500     \expandafter\convertcolorspec#2\HyColor@model@rgb#2%
501     \expandafter\HyColor@NormalizeCommaRGB#2\@nil#2%
502     }{%
503     \let#2\relax
504     \HyColor@ErrorSpecNoXcolor{#3}{#4}%
505     }%
506     }%
507 \fi
508 }%
509 }

```

2.6 Package `attachfile2`

Before PDF-1.7 only RGB values are permitted in annotations. Since PDF-1.7 the color entry in annotations understands several color models, depending on the size of the color array:

- Zero entries: means transparent, not useful for file attachments. AR7/Linux and AR8/Linux show black instead.
- One entry: color model ‘gray’.
- Three entries: color model ‘rgb’.
- Four entries: color model ‘cmyk’.

An empty color specification is interpreted as “no color”.

`\HyColor@DetectPdfVersion`

```

510 \def\HyColor@DetectPdfVersion{%
511     \begingroup\expandafter\expandafter\expandafter\endgroup
512     \expandafter\ifx\curname Hy@pdfversion\endcurname\relax

```

```

513 \global\chardef\HyColor@PdfVersion=0 %
514 \else
515 \global\chardef\HyColor@PdfVersion=\Hy@pdfversion\relax
516 \fi
517 \global\let\HyColor@DetectPdfVersion\relax
518 }

```

\HyColor@SpaceToComma

```

519 \def\HyColor@SpaceToComma#1 #2\@nil{%
520 #1%
521 \ifx\relax#2\relax
522 \expandafter\@gobble
523 \else
524 ,%
525 \expandafter\@firstofone
526 \fi
527 {%
528 \HyColor@SpaceToComma#2\@nil
529 }%
530 }%

```

\HyColor@AttachfileColor

```

531 \def\HyColor@AttachfileColor#1#2#3#4#5#6{%
532 \def#2{#1}%
533 \ifx#2\@empty
534 \let#3\@gobble
535 \let#4\@empty
536 \else
537 \HyColor@resultfalse
538 \HyColor@XZeroOneThreeFour{#1}#3{#5}{#6}%
539 \ifHyColor@result
540 \edef#2{%
541 [rgb]{\expandafter\HyColor@SpaceToComma#3 \@nil}%
542 }%
543 \fi
544 \ifx\HyColor@model\HyColor@model@rgb
545 \edef#4{/C[#3]}% hash-ok
546 \edef#3##1{%
547 #3 %
548 \noexpand\csname atfi@SETRGBCOLOR##1\noexpand\endcsname
549 }%
550 \else
551 \ifx\HyColor@model\HyColor@model@gray
552 \HyColor@DetectPdfVersion
553 \ifnum\HyColor@PdfVersion<7 %
554 \edef#4{/C[#3 #3 #3]}% hash-ok
555 \else
556 \edef#4{/C[#3]}% hash-ok
557 \fi
558 \edef#3##1{%
559 #3 %
560 \noexpand\csname atfi@SETGRAYCOLOR##1\noexpand\endcsname
561 }%
562 \else
563 \ifx\HyColor@model\HyColor@model@cmyk
564 \HyColor@DetectPdfVersion
565 \ifnum\HyColor@PdfVersion<7 %
566 \HyColor@IfModel{#1}{%

```

```

567         \HyColor@IfXcolor{%
568             \convertcolorspec\HyColor@model\HyColor@values
569                 \HyColor@model@rgb#4%
570             \expandafter\HyColor@NormalizeCommaRGB#4\@nil#4%
571             \edef#4{/C[#4]}% hash-ok
572         }{%
573             \let#4\@empty
574             \HyColor@ErrorModelNoXcolor{#5}{#6}%
575         }%
576     }{%
577         \HyColor@IfXcolor{%
578             \extractcolorspec{#1}#4%
579             \expandafter\convertcolorspec#4%
580                 \HyColor@model@rgb#4%
581             \expandafter\HyColor@NormalizeCommaRGB#4\@nil#4%
582             \edef#4{/C[#4]}% hash-ok
583         }{%
584             \let#4\@empty
585             \HyColor@ErrorSpecNoXcolor{#5}{#6}%
586         }%
587     }%
588 \else
589     \edef#4{/C[#3]}% hash-ok
590 \fi
591 \edef#3##1{%
592     #3 %
593     \noexpand\cename atfi@SETCMYKCOLOR##1\noexpand\endcename
594 }%
595 \else
596 \ifx\HyColor@model\HyColor@model@empty
597 \PackageError{#5}{%
598     Color model 'empty' is not permitted for option '#6'%
599 }@ehc
600 \let#2\@empty
601 \let#3\@gobble
602 \let#4\@empty
603 \else
604 \ifx\HyColor@model\relax % (missing xcolor)
605 \let#3\@gobble
606 \let#4\@empty
607 \else
608 \PackageError{#5}{%
609     Internal error: unsupported color model%
610 }@ehc
611 \fi
612 \fi
613 \fi
614 \fi
615 \fi
616 \fi
617 }
618 </package>

```

2.7 Patch for package xcolor

Because the test files triggered a bug in package xcolor of version 2007/01/21 v2.11. I contacted the author of xcolor Uwe Kern. He responded with a test

version 2007/03/27 v2.12a00 that fixes the problem. However, apparently he did not find the time for an official release yet. Thus I have reluctantly written the following patch package using the fixes of v2.12a00.

The patch is immediately applied if package `xcolor` is already loaded. Otherwise the patch is delayed using `\AfterPackage` if package `scrfile` is loaded. As last resort `\AtBeginDocument` is used.

```
619 (*xcolor)
620 \NeedsTeXFormat{LaTeX2e}
621 \ProvidesPackage{xcolor-patch}[2020/01/27 v1.10 xcolor patch]
622 \RequirePackage{hopatch}
623 \hopatch@AfterPackage{xcolor}{%
```

`\XC@ifxcase`

```
624 \long\def\reserved@a#1#2#3{%
625 \long\def\@tmp##1##2{%
626 \ifx##1#1%
627 \toks@{##2}%
628 \expandafter\remove@to@nnil
629 \else
630 \expandafter\@tmp
631 \fi
632 }%
633 \@tmp#2#1{#3}\@nnil\the\toks@
634 }%
635 \ifx\CX@ifxcase\reserved@a
636 \long\def\CX@ifxcase#1#2#3{%
637 \long\def\CX@if@##1##2{%
638 \ifx##1#1%
639 \toks@{##2}%
640 \expandafter\remove@to@nnil
641 \else
642 \expandafter\CX@if@
643 \fi
644 }%
645 \CX@if@#2#1{#3}\@nnil
646 \the\toks@
647 }%
648 \fi
```

`\XC@ifcase`

```
649 \long\def\reserved@a#1#2#3{%
650 \long\def\@tmp##1##2{%
651 \@expandtwoargs\in@{,#1,}{,##1,}%
652 \ifin@
653 \toks@{##2}%
654 \expandafter\remove@to@nnil
655 \else
656 \expandafter\@tmp
657 \fi
658 }%
659 \@tmp#2{#1}{#3}\@nnil
660 \the\toks@
661 }%
662 \ifx\CX@ifcase\reserved@a
663 \long\def\CX@ifcase#1#2#3{%
664 \long\def\CX@if@##1##2{%
665 \@expandtwoargs\in@{,#1,}{,##1,}%
```

```

666     \ifin@
667         \toks@{##2}%
668         \expandafter\remove@to@nnil
669     \else
670         \expandafter\XC@if@
671     \fi
672 }%
673 \XC@if@#2{#1}{#3}\@nnil
674 \the\toks@
675 }%
676 \fi

```

\XC@cnv@gray

```

677 \def\reserved@a#1,{%
678 \XC@ifxcase\tm{%
679 \XC@mod@rgb{%
680 \XC@calcN{#1}\@tmp
681 \edef\@tmp{\@tmp,\@tmp,\@tmp}%
682 }%
683 \XC@mod@cmy{%
684 \XC@calcC{#1}\@tmp
685 \edef\@tmp{\@tmp,\@tmp,\@tmp}%
686 }%
687 \XC@mod@cmyk{%
688 \XC@calcC{#1}\@tmp
689 \edef\@tmp{0,0,0,\@tmp}%
690 }%
691 \XC@mod@RGB{%
692 \edef\@scl{\rangeRGB}%
693 \XC@calcM{#1}\@tmp
694 \edef\@tmp{\@tmp,\@tmp,\@tmp}%
695 }%
696 \XC@mod@HTML{%
697 \edef\@scl{\@cclv}%
698 \XC@calcM{#1}\@tmp
699 \XC@calcH\@tmp\@tmp
700 \edef\@tmp{\@tmp\@tmp\@tmp}%
701 }%
702 \XC@mod@HSB{%
703 \edef\@scl{\rangeHSB}%
704 \XC@calcM{#1}\@tmp
705 \edef\@tmp{0,0,\@tmp}%
706 }%
707 \XC@mod@Gray{%
708 \edef\@scl{\rangeGray}%
709 \XC@calcM{#1}\@tmp
710 }%
711 }%
712 {%
713 \XC@calcN{#1}\@tmp
714 \edef\@tmp{0,0,\@tmp}%
715 }%
716 }%
717 \ifx\XC@cnv@gray\reserved@a
718 \def\XC@cnv@gray#1,{%
719 \XC@ifxcase\tm{%
720 \XC@mod@rgb{%
721 \XC@calcN{#1}\@tmp

```

```

722     \edef\@@tmp{\@@tmp,\@@tmp,\@@tmp}%
723 }%
724 \XC@mod@gray{%
725 \XC@mod@cmy{%
726   \XC@calcC{#1}\@@tmp
727   \edef\@@tmp{\@@tmp,\@@tmp,\@@tmp}%
728 }%
729 \XC@mod@cmyk{%
730   \XC@calcC{#1}\@@tmp
731   \edef\@@tmp{0,0,0,\@@tmp}%
732 }%
733 \XC@mod@RGB{%
734   \edef\@@scl{\rangeRGB}%
735   \XC@calcM{#1}\@@tmp
736   \edef\@@tmp{\@@tmp,\@@tmp,\@@tmp}%
737 }%
738 \XC@mod@HTML{%
739   \edef\@@scl{\cc1v}%
740   \XC@calcM{#1}\@@tmp
741   \XC@calcH\@@tmp\@@tmp
742   \edef\@@tmp{\@@tmp\@@tmp\@@tmp}%
743 }%
744 \XC@mod@HSB{%
745   \edef\@@scl{\rangeHSB}%
746   \XC@calcM{#1}\@@tmp
747   \edef\@@tmp{0,0,\@@tmp}%
748 }%
749 \XC@mod@Gray{%
750   \edef\@@scl{\rangeGray}%
751   \XC@calcM{#1}\@@tmp
752 }%
753 }%
754 {%
755   \XC@calcN{#1}\@@tmp
756   \edef\@@tmp{0,0,\@@tmp}%
757 }%
758 }%
759 \fi

```

2.7.1 Fix fragile \@frameb@x

\fbox becomes fragile, because the internal \@frameb@x is redefined by package xcolor. The redefinition is no longer robust. Test file:

```

\documentclass{article}
\usepackage{xcolor}
\makeatletter
\protected@edef\x{\fbox{abc}}
\@end

760 \@ifundefined{XC@frameb@x}{%
761   \expandafter\let\csname XC@frameb@x \endcsname\XC@frameb@x
762   \edef\XC@frameb@x{%
763     \noexpand\protect
764     \expandafter\noexpand\csname XC@frameb@x \endcsname
765   }%
766   \expandafter\ifx\csname XC@frameb@x \endcsname\@frameb@x
767   \let\@frameb@x\XC@frameb@x
768 }%
\fi

```

```
769 }{}%
770 }
771 </xcolor>
```

3 Installation

3.1 Download

Package. This package is available on CTAN¹:

[CTAN:macros/latex/contrib/hycolor/hycolor.dtx](#) The source file.

[CTAN:macros/latex/contrib/hycolor/hycolor.pdf](#) Documentation.

3.2 Package installation

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

```
tex hycolor.dtx
```

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

```
hycolor.sty      → tex/latex/hycolor/hycolor.sty
xcolor-patch.sty → tex/latex/hycolor/xcolor-patch.sty
hycolor.pdf      → doc/latex/hycolor/hycolor.pdf
hycolor.dtx      → source/latex/hycolor/hycolor.dtx
```

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`.

3.3 Refresh file name databases

If your \TeX distribution (\TeX Live, MiK \TeX , ...) relies on file name databases, you must refresh these. For example, \TeX Live users run `texhash` or `mktextlsr`.

3.4 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{hycolor.dtx}
```

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

¹[CTAN:pkg/hycolor](#)

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

4 History

[2007/04/09 v1.0]

- First version.

[2007/04/11 v1.1]

- Line ends sanitized.

[2008/07/29 v1.2]

- Support for package `attachfile2` added.

[2008/08/01 v1.3]

- Patch package `xcolor-patch` added that fixes bugs in package `xcolor` to get the test files running.

[2008/09/08 v1.4]

- Fix added to package `xcolor-patch`: Fragile `\@frameb@x` (used in `\fbox`) is made robust.

[2009/10/02 v1.5]

- Doku fixes (Herbert Voss).

[2009/12/12 v1.6]

- Short info shortened.

[2011/01/30 v1.7]

- Package `xcolor-patch` uses package `hopatch`.

[2016/05/16 v1.8]

- Documentation updates.

[2019/12/15 v1.9]

- Documentation updates.
- Do not load xcolor-patch by default.

[2020-01-27 v1.10]

- extra expansion step to avoid
! LaTeX Error: Undefined color ‘[rgb]’ errors in hyperref.
- add fix to Gray model from xcolor-patch.

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	
<code>\!</code>	93
<code>\"</code>	98
<code>\+</code>	96
<code>\-</code>	95
<code>\.</code>	451
<code>\:</code>	94
<code>\;</code>	97
<code>\></code>	99
<code>\@@scl</code>	22, 27, 33, 38, 64, 69, 75, 80, 692, 697, 703, 708, 734, 739, 745, 750
<code>\@tmp</code> ...	10, 11, 14, 15, 18, 19, 23, 24, 28, 29, 30, 34, 35, 39, 43, 44, 51, 52, 56, 57, 60, 61, 65, 66, 70, 71, 72, 76, 77, 81, 85, 86, 625, 630, 633, 650, 656, 659, 680, 681, 684, 685, 688, 689, 693, 694, 698, 699, 700, 704, 705, 709, 713, 714, 721, 722, 726, 727, 730, 731, 735, 736, 740, 741, 742, 746, 747, 751, 755, 756
<code>\@ReturnAfterFi</code>	151, 156, <u>249</u>
<code>\@car</code>	436
<code>\@cclv</code>	27, 69, 697, 739
<code>\@ehc</code>	222, 229, 385, 599, 610
<code>\@empty</code>	126, 136, 137, 197, 204, 211, 269, 271, 291, 299, 330, 371, 374, 376, 433, 492, 533, 535, 573, 584, 600, 602, 606
<code>\@expandtwoargs</code>	651, 665
<code>\@firstofone</code>	525
<code>\@firstoftwo</code>	242, 255, 338, 393, 423, 463
<code>\@frameb@x</code>	766, 767
<code>\@gobble</code>	522, 534, 601, 605
<code>\@ifnextchar</code>	232, 278
<code>\@ifundefined</code>	760
<code>\@ne</code>	131
<code>\@nil</code>	127, 130, 135, 140, 147, 152, 164, 172, 187, 190, 209, 237, 239, 244, 273, 280, 283, 304, 306, 321, 334, 345, 348, 358, 401, 406, 436, 475, 478, 496, 501, 519, 528, 541, 570, 581
<code>\@nnil</code>	633, 645, 659, 673
<code>\@onelevel@sanitize</code>	118, 259, 261, 263, 265, 267
<code>\@secondoftwo</code>	247, 253, 339, 395, 425, 466
<code>\@tempa</code>	7, 47, 90
<code>\@</code>	132, 148
A	
<code>\AtBeginDocument</code>	6
C	
<code>\catcode</code>	93, 94, 95, 96, 97, 98, 99
<code>\chardef</code>	513, 515
<code>\color</code>	281, 284
<code>\convertcolorspec</code> .	185, 208, 309, 319, 350, 356, 473, 500, 568, 579
<code>\csname</code>	112, 252, 392, 512, 548, 560, 593, 761, 764, 766
E	
<code>\endcsname</code>	112, 252, 392, 512, 548, 560, 593, 761, 764, 766
<code>\extractcolorspec</code> .	207, 337, 499, 578
H	
<code>\hopatch@AfterPackage</code>	623
<code>\Hy@pdfversion</code>	515
<code>\HyColor@@@UseColor</code>	278, 283
<code>\HyColor@@UseColor</code>	278, 280
<code>\HyColor@@IfRGB</code>	401, 406
<code>\HyColor@@UseColor</code>	273, 277
<code>\HyColor@AttachfileColor</code>	<u>531</u>

T		\XC@if@ ... 637, 642, 645, 664, 670, 673	
\the	633, 646, 660, 674	\XC@ifcase 649	
\tm	8, 49, 678, 719	\XC@ifxcase 8, 49, 624, 678, 719	
\toks@	627,	\XC@mod@cmy 13, 55, 683, 725	
	633, 639, 646, 653, 660, 667, 674	\XC@mod@cmyk 17, 59, 687, 729	
X		\XC@mod@Gray 37, 79, 707, 749	
\x	100, 114, 117, 121, 399, 404	\XC@mod@gray 54, 724	
\XC@calcC		\XC@mod@HSB 32, 74, 702, 744	
	14, 18, 56, 60, 684, 688, 726, 730	\XC@mod@HTML 26, 68, 696, 738	
\XC@calcH	29, 71, 699, 741	\XC@mod@RGB 21, 63, 691, 733	
\XC@calcM	23,	\XC@mod@rgb 9, 50, 679, 720	
	28, 34, 39, 65, 70, 76, 81, 693,	Y	
	698, 704, 709, 735, 740, 746, 751	\y	109, 121
\XC@calcN		Z	
	10, 43, 51, 85, 680, 713, 721, 755	\z@	123, 455
\XC@cnv@gray	47, 48, 677	\zap@space	126
\XC@frameb@x	761, 762, 767		