

# The `hycolor` package

Heiko Oberdiek\*

2016/05/16 v1.8

## 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

<b>1</b>	<b>Documentation</b>	<b>2</b>
1.1	Summary	2
<b>2</b>	<b>Implementation</b>	<b>3</b>
2.1	Normalization	3
2.1.1	Sanitize value of color option	3
2.1.2	Normalize result	4
2.2	Main algorithm for color options	5
2.3	Package <code>bookmark</code>	6
2.4	Utils	7
2.5	Package <code>hyperref</code>	8
2.5.1	Options <code>Hyp.*color</code>	8
2.5.2	Generic algorithm	11
2.5.3	Field options	13
2.5.4	Detection for naked RGB values	13
2.5.5	Options <code>*bordercolor</code>	15
2.6	Package <code>attachfile2</code>	16
2.7	Patch for package <code>xcolor</code>	19
2.7.1	Fix fragile <code>\@frameb@x</code>	21
<b>3</b>	<b>Test</b>	<b>22</b>
3.1	Test for package <code>attachfile2</code>	26
3.2	Test for package <code>xcolor</code>	29
3.2.1	Test for <code>\@frameb@x/\fbox</code>	30
<b>4</b>	<b>Installation</b>	<b>30</b>
4.1	Download	30
4.2	Bundle installation	30
4.3	Package installation	30
4.4	Refresh file name databases	31
4.5	Some details for the interested	31

---

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

<b>5 History</b>	<b>31</b>
[2007/04/09 v1.0]	31
[2007/04/11 v1.1]	32
[2008/07/29 v1.2]	32
[2008/08/01 v1.3]	32
[2008/09/08 v1.4]	32
[2009/10/02 v1.5]	32
[2009/12/12 v1.6]	32
[2011/01/30 v1.7]	32
[2016/05/16 v1.8]	32

# 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
BKM.color	gray, rgb	no	no
Hyp.*color	all	no	no
Hyp.*bordercolor	gray, rgb	yes	no
Field.*color	gray, rgb, cmyk	yes	yes
AtFi.color	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 \hypersetup
Field	Package hyperref: Form field options
AtFi	Package attachfile2: option color

## 2 Implementation

```

1 <*package>
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{hycolor}%
4   [2016/05/16 v1.8 Color options for hyperref/bookmark (H0)]%
5 \RequirePackage{xcolor-patch}[2016/05/16]
```

### 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

```

6 \begingroup
7   \catcode`\!=13 %
8   \catcode`\:=13 %
9   \catcode`\-=13 %
10  \catcode`\+=13 %
11  \catcode`\;=13 %
12  \catcode`\ "=13 %
13  \catcode`\>=13 %
14  \edef\x{%
15    \def\noexpand!\{\string!}%
16    \def\noexpand:\{\string:}%
17    \def\noexpand-{\string-}%
18    \def\noexpand+{\string+}%
19    \def\noexpand;\{\string;}%
20    \def\noexpand"\{\string"}%
21    \def\noexpand>\{\string>}%
22  }%
23  \def\y#1{\endgroup
24    \def\HyColor@DefSanitized##1##2{%
25      \begingroup
26        \csname @safe@activetrue\endcsname
```

```

27      #1%
28      \edef\x{\endgroup
29      \def\noexpand##1{##2}%
30      }%
31      \x
32      \@onelevel@sanitize##1%
33      }%
34      }%
35 \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

```

36 \def\HyColor@NormalizeNum#1#2{%
37   \ifdim#1pt<\z@
38     \def#2{0}%
39   \else
40     \edef#2{\zap@space#1 \@empty}%
41     \expandafter\HyColor@CheckDot#2..\@nil#2%
42   \fi
43 }
44 \def\HyColor@CheckDot#1.#2.#3\@nil#4{%
45   \ifnum0#1<\@ne
46     \ifx\#2\%
47       \def#4{0}%
48     \else
49       \edef#4{\HyColor@ReverseString#2\@nil{}}%
50       \edef#4{\expandafter\HyColor@StripLeadingZeros#4\@empty}%
51       \ifx#4\@empty
52         \def#4{0}%
53       \else
54         \edef#4{.\expandafter\HyColor@ReverseString#4\@nil{}}%
55       \fi
56     \fi
57   \else
58     \def#4{1}%

```

```

59 \fi
60 }
61 \def\HyColor@ReverseString#1#2\@nil#3{%
62 \ifx\#2\%
63 #1#3%
64 \else
65 \@ReturnAfterFi{%
66 \HyColor@ReverseString#2\@nil{#1#3}%
67 }%
68 \fi
69 }
70 \long\def\@ReturnAfterFi#1\fi{\fi#1}
71 \def\HyColor@StripLeadingZeros#1{%
72 \ifx#10%
73 \expandafter\HyColor@StripLeadingZeros
74 \else
75 #1%
76 \fi
77 }

```

\HyColor@NormalizeCommaRGB

```

78 \def\HyColor@NormalizeCommaRGB#1,#2,#3\@nil#4{%
79 \HyColor@NormalizeNum{#1}\HyColor@temp
80 \let#4\HyColor@temp
81 \HyColor@NormalizeNum{#2}\HyColor@temp
82 \edef#4{#4 \HyColor@temp}%
83 \HyColor@NormalizeNum{#3}\HyColor@temp
84 \edef#4{#4 \HyColor@temp}%
85 }

```

\HyColor@NormalizeCommaCMYK

```

86 \def\HyColor@NormalizeCommaCMYK#1,#2,#3,#4\@nil#5{%
87 \HyColor@NormalizeNum{#1}\HyColor@temp
88 \let#5\HyColor@temp
89 \HyColor@NormalizeNum{#2}\HyColor@temp
90 \edef#5{#5 \HyColor@temp}%
91 \HyColor@NormalizeNum{#3}\HyColor@temp
92 \edef#5{#5 \HyColor@temp}%
93 \HyColor@NormalizeNum{#4}\HyColor@temp
94 \edef#5{#5 \HyColor@temp}%
95 }

```

## 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
end
```

---

```
96 \def\HyColor@BookmarkColor#1#2#3#4{%
97   \HyColor@IfModel{#1}-{%
98     \HyColor@IfXcolor{%
99       \convertcolorspec\HyColor@model\HyColor@values
100         \HyColor@model@rgb#2%
101       \expandafter\HyColor@NormalizeCommaRGB#2\@nil#2%
102     }{%
103       \ifx\HyColor@model\HyColor@model@rgb
104         \expandafter\HyColor@NormalizeCommaRGB\HyColor@values\@nil#2%
105       \else
106         \ifx\HyColor@model\HyColor@model@gray
107           \expandafter\HyColor@NormalizeNum
108           \expandafter{\HyColor@values}#2%
109         \edef#2{#2 #2 #2}%

```

```

110         \else
111         \let#2\@empty
112         \HyColor@ErrorModelNoXcolor{#3}{#4}%
113     \fi
114 \fi
115 }%
116 }{%
117     \let#2\HyColor@values
118     \ifx#2\@empty
119     \else
120     \HyColor@IfXcolor{%
121         \extractcolorspec{#1}#2%
122         \expandafter\convertcolorspec#2\HyColor@model@rgb#2%
123         \expandafter\HyColor@NormalizeCommaRGB#2\@nil#2%
124     }{%
125         \let#2\@empty
126         \HyColor@ErrorSpecNoXcolor{#3}{#4}%
127     }%
128     \fi
129 }%
130 }

131 \def\HyColor@ErrorModelNoXcolor#1#2{%
132     \PackageError{#1}{%
133         Color model '\HyColor@model' is not supported\MessageBreak
134         without package `xcolor' in\MessageBreak
135         `#2=[\HyColor@model]{\HyColor@values}'%
136     }{\@ehc
137 }

138 \def\HyColor@ErrorSpecNoXcolor#1#2{%
139     \PackageError{#1}{%
140         This color specification is not supported\MessageBreak
141         without package `xcolor' in\MessageBreak
142         `#2=\HyColor@values'%
143     }{\@ehc
144 }

145 \def\HyColor@IfModel#1{%
146     \@ifnextchar[{%
147         \HyColor@WithModel
148     }{%
149         \HyColor@WithoutModel
150     }%
151     #1\@nil
152 }

153 \def\HyColor@WithModel[#1]#2\@nil{%
154     \HyColor@DefSanitized\HyColor@model{#1}%
155     \HyColor@DefSanitized\HyColor@values{#2}%
156     \@firstoftwo
157 }

158 \def\HyColor@WithoutModel#1\@nil{%
159     \let\HyColor@model\relax
160     \HyColor@DefSanitized\HyColor@values{#1}%
161     \@secondoftwo
162 }

```

## 2.4 Utils

\@ReturnAfterFi

```

163 \long\def\@ReturnAfterFi#1\fi{\fi#1}

\HyColor@IfXcolor

164 \def\HyColor@IfXcolor{%
165   \begingroup\expandafter\expandafter\expandafter\endgroup
166   \expandafter\ifx\csname convertcolorspec\endcsname\relax
167     \expandafter\@secondoftwo
168   \else
169     \expandafter\@firstoftwo
170   \fi
171 }

172 \def\HyColor@model@empty{empty}
173 \@onelevel@sanitize\HyColor@model@empty
174 \def\HyColor@model@gray{gray}
175 \@onelevel@sanitize\HyColor@model@gray
176 \def\HyColor@model@rgb{rgb}
177 \@onelevel@sanitize\HyColor@model@rgb
178 \def\HyColor@model@cmyk{cmyk}
179 \@onelevel@sanitize\HyColor@model@cmyk
180 \def\HyColor@model@Gray{Gray}
181 \@onelevel@sanitize\HyColor@model@Gray

```

## 2.5 Package hyperref

### 2.5.1 Options Hyp.\*color

```

182 \def\HyColor@UseColor#1{%
183   \ifx#1\relax
184   \else
185     \ifx#1\@empty
186     \else
187       \expandafter\HyColor@@UseColor#1\@nil
188     \fi
189   \fi
190 }
191 \def\HyColor@@UseColor{%
192   \@ifnextchar[\HyColor@@@UseColor\HyColor@@@UseColor
193 }
194 \def\HyColor@@@UseColor[#1]#2\@nil{%
195   \color[{#1}]{#2}%
196 }
197 \def\HyColor@@@UseColor#1\@nil{%
198   \color{#1}%
199 }

```



---

**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
```

---

```
200 \def\HyColor@HyperrefColor#1#2{%
201   \HyColor@IfModel{#1}{%
202     \edef#2{[\HyColor@model]}\HyColor@values}}%
203 }{%
204   \let#2\HyColor@values
205   \ifx#2\@empty
206     \let#2\relax
207   \fi
208 }%
209 }
```



### 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
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

```
210 \def\HyColor@XZeroOneThreeFour#1#2#3#4{%
211   \HyColor@IfModel{#1}{%
212     \ifx\HyColor@model\HyColor@model@empty
213       \let#2\@empty
214     \else\ifx\HyColor@model\HyColor@model@gray
215       \expandafter\HyColor@NormalizeNum
216       \expandafter{\HyColor@values}#2%
217     \else\ifx\HyColor@model\HyColor@model@rgb
218       \expandafter\HyColor@NormalizeCommaRGB\HyColor@values\@nil#2%
219     \else\ifx\HyColor@model\HyColor@model@cmyk
220       \expandafter\HyColor@NormalizeCommaCMYK\HyColor@values\@nil#2%
221     \else\ifx\HyColor@model\HyColor@model@Gray
222       \HyColor@IfXcolor{%
223         \convertcolorspec\HyColor@model\HyColor@values
224         \HyColor@model@gray#2%
225       \expandafter\HyColor@NormalizeNum\expandafter{#2}#2%
226       \let\HyColor@model\HyColor@model@gray
227     }{%
228       \let#2\relax
229     \HyColor@ErrorModelNoXcolor{#3}{#4}%
230   }%
231 \else
232   \HyColor@IfXcolor{%
233     \convertcolorspec\HyColor@model\HyColor@values
234     \HyColor@model@rgb#2%
235     \expandafter\HyColor@NormalizeCommaRGB#2\@nil#2%
236     \let\HyColor@model\HyColor@model@rgb
237   }{%
238     \let#2\relax
239   \HyColor@ErrorModelNoXcolor{#3}{#4}%
240 }%
241 \fi\fi\fi\fi\fi
242 }{%
243   \let#2\HyColor@values
244   \ifx#2\@empty
245     \let#2\relax
246   \else
247     \expandafter\HyColor@IfRGB\expandafter{\HyColor@values}{%
248       \expandafter\HyColor@NormalizeCommaRGB\HyColor@values\@nil#2%
249     }{%
250       \HyColor@IfXcolor{%
251         \expandafter\extractcolorspec\expandafter{\HyColor@values}#2%
252         \edef\HyColor@model{\expandafter\@firstoftwo#2}%
253         \edef\HyColor@values{\expandafter\@secondoftwo#2}%
254         \ifx\HyColor@model\HyColor@model@gray
255           \expandafter\HyColor@NormalizeNum\expandafter
256           {\HyColor@values}#2%
257         \else\ifx\HyColor@model\HyColor@model@rgb
258           \expandafter\HyColor@NormalizeCommaRGB
259           \HyColor@values\@nil#2%
260         \else\ifx\HyColor@model\HyColor@model@cmyk
261           \expandafter\HyColor@NormalizeCommaCMYK
262           \HyColor@values\@nil#2%
263         \else\ifx\HyColor@model\HyColor@model@Gray
264           \convertcolorspec\HyColor@model\HyColor@values
265           \HyColor@model@gray#2%
266         \expandafter\HyColor@NormalizeNum\expandafter
```

```

267         {\HyColor@values}#2%
268         \let\HyColor@model\HyColor@model@gray
269     \else
270         \convertcolorspec\HyColor@model\HyColor@values
271         \HyColor@model@rgb#2%
272         \expandafter\HyColor@NormalizeCommaRGB#2\@nil#2%
273         \let\HyColor@model\HyColor@model@rgb
274     \fi\fi\fi\fi
275 }{%
276     \let#2\relax
277     \HyColor@ErrorSpecNoXcolor{#3}{#4}%
278 }%
279 }%
280 \fi
281 }%
282 }

```

### 2.5.3 Field options

\HyColor@FieldBColor

```

283 \let\HyColor@FieldBColor\HyColor@XZeroOneThreeFour

```

\HyColor@FieldColor

```

284 \def\HyColor@FieldColor#1#2#3#4{%
285     \let\HyColor@model\@empty
286     \HyColor@XZeroOneThreeFour{#1}{#2}{#3}{#4}%
287     \ifx#2\relax
288         \let#2\@empty
289     \else
290         \ifx#2\@empty
291         \else
292             \ifx\HyColor@model\HyColor@model@gray
293                 \edef#2{#2 g}%
294             \else\ifx\HyColor@model\HyColor@model@rgb
295                 \edef#2{#2 rg}%
296             \else\ifx\HyColor@model\HyColor@model@cmyk
297                 \edef#2{#2 k}%
298             \else
299                 \PackageError{#3}{Internal error: unsupported color model}\@ehc
300             \fi\fi\fi
301         \fi
302     \fi
303 }

```

### 2.5.4 Detection for naked RGB values

\HyColor@IfRGB

```

304 \newif\ifHyColor@result
305 \begingroup\expandafter\expandafter\expandafter\endgroup
306 \expandafter\ifx\csname pdfmatch\endcsname\relax
307     \expandafter\@firstoftwo
308 \else
309     \expandafter\@secondoftwo
310 \fi
311 {%
312     \begingroup
313         \def\x#1{\endgroup
314             \def\HyColor@IfRGB##1{%

```

```

315         \HyColor@@IfRGB##1#1#1#1\@nil
316     }%
317 }%
318 \x{ }%
319 \edef\HyColor@TwoSpaces{\space\space}%
320 \def\HyColor@@IfRGB#1 #2 #3 #4\@nil{%
321     \HyColor@resulttrue
322     \def\HyColor@temp{#4}%
323     \ifx\HyColor@temp\HyColor@TwoSpaces
324         \HyColor@CheckNum{#1}%
325         \ifHyColor@result
326             \HyColor@CheckNum{#2}%
327             \ifHyColor@result
328                 \HyColor@CheckNum{#3}%
329             \fi
330         \fi
331     \else
332         \HyColor@resultfalse
333     \fi
334     \ifHyColor@result
335         \let\HyColor@model\HyColor@model@rgb
336         \edef\HyColor@values{#1,#2,#3}%
337         \expandafter\@firstoftwo
338     \else
339         \expandafter\@secondoftwo
340     \fi
341 }%
342 \def\HyColor@zero{0}%
343 \def\HyColor@one{1}%
344 \def\HyColor@dot{.}%
345 \def\HyColor@CheckNum#1{%
346     \def\HyColor@temp{#1}%
347     \ifx\HyColor@temp\@empty
348         \HyColor@resultfalse
349     \else
350         \edef\HyColor@temp{\@car#1\@nil}%
351         \ifx\HyColor@temp\HyColor@zero
352             \else
353                 \ifx\HyColor@temp\HyColor@one
354                     \else
355                         \ifx\HyColor@temp\HyColor@dot
356                             \else
357                                 \HyColor@resultfalse
358                             \fi
359                         \fi
360                     \fi
361                 \fi
362             }%
363 }{%
364     \def\HyColor@MatchNum{%
365         (0*1|string\.0*|0*1|0+\string\.[0-9]*|\string\.[0-9]+)%
366     }%
367     \def\HyColor@IfRGB#1{%
368         \ifnum\pdfmatch{~\HyColor@MatchNum\space\HyColor@MatchNum
369             \space\HyColor@MatchNum$}{#1}>\z@
370             \let\HyColor@model\HyColor@model@rgb
371             \edef\HyColor@values{%
372                 \expandafter\strip@prefix\pdfmatch1,%

```

```

373     \expandafter\strip@prefix\pdflastmatch2,%
374     \expandafter\strip@prefix\pdflastmatch3%
375   }%
376   \HyColor@resulttrue
377   \expandafter\@firstoftwo
378 \else
379   \HyColor@resultfalse
380   \expandafter\@secondoftwo
381 \fi
382 }%
383 }

```

### 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

```

384 \def\HyColor@HyperrefBorderColor#1#2#3#4{%

```

```

385 \HyColor@IfModel{#1}{%
386   \HyColor@IfXcolor{%
387     \convertcolorspec\HyColor@model\HyColor@values
388       \HyColor@model@rgb#2%
389     \expandafter\HyColor@NormalizeCommaRGB#2\@nil#2%
390   }{%
391     \ifx\HyColor@model\HyColor@model@rgb
392       \expandafter\HyColor@NormalizeCommaRGB\HyColor@values\@nil#2%
393     \else
394       \ifx\HyColor@model\HyColor@model@gray
395         \expandafter\HyColor@NormalizeNum
396           \expandafter{\HyColor@values}#2%
397         \edef#2{#2 #2 #2}%
398       \else
399         \let#2\relax
400         \HyColor@ErrorModelNoXcolor{#3}{#4}%
401       \fi
402     \fi
403   }%
404 }{%
405   \let#2\HyColor@values
406   \ifx#2\@empty
407     \let#2\relax
408   \else
409     \expandafter\HyColor@IfRGB\expandafter{\HyColor@values}{%
410       \expandafter\HyColor@NormalizeCommaRGB\HyColor@values\@nil#2%
411     }{%
412       \HyColor@IfXcolor{%
413         \extractcolorspec{#1}#2%
414         \expandafter\convertcolorspec#2\HyColor@model@rgb#2%
415         \expandafter\HyColor@NormalizeCommaRGB#2\@nil#2%
416       }{%
417         \let#2\relax
418         \HyColor@ErrorSpecNoXcolor{#3}{#4}%
419       }%
420     }%
421   \fi
422 }%
423 }

```

## 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

```

424 \def\HyColor@DetectPdfVersion{%

```



```

425 \begingroup\expandafter\expandafter\expandafter\endgroup
426 \expandafter\ifx\csname Hy@pdfversion\endcsname\relax
427 \global\chardef\HyColor@PdfVersion=0 %
428 \else
429 \global\chardef\HyColor@PdfVersion=\Hy@pdfversion\relax
430 \fi
431 \global\let\HyColor@DetectPdfVersion\relax
432 }

```

\HyColor@SpaceToComma

```

433 \def\HyColor@SpaceToComma#1 #2\@nil{%
434 #1%
435 \ifx\relax#2\relax
436 \expandafter\@gobble
437 \else
438 ,%
439 \expandafter\@firstofone
440 \fi
441 {%
442 \HyColor@SpaceToComma#2\@nil
443 }%
444 }%

```

\HyColor@AttachfileColor

```

445 \def\HyColor@AttachfileColor#1#2#3#4#5#6{%
446 \def#2{#1}%
447 \ifx#2\@empty
448 \let#3\@gobble
449 \let#4\@empty
450 \else
451 \HyColor@resultfalse
452 \HyColor@XZeroOneThreeFour{#1}#3{#5}{#6}%
453 \ifHyColor@result
454 \edef#2{%
455 [rgb]{\expandafter\HyColor@SpaceToComma#3 \@nil}%
456 }%
457 \fi
458 \ifx\HyColor@model\HyColor@model@rgb
459 \edef#4{/C[#3]}% hash-ok
460 \edef#3##1{%
461 #3 %
462 \noexpand\csname atfi@SETRGBCOLOR##1\noexpand\endcsname
463 }%
464 \else
465 \ifx\HyColor@model\HyColor@model@gray
466 \HyColor@DetectPdfVersion
467 \ifnum\HyColor@PdfVersion<7 %
468 \edef#4{/C[#3 #3 #3]}% hash-ok
469 \else
470 \edef#4{/C[#3]}% hash-ok
471 \fi
472 \edef#3##1{%
473 #3 %
474 \noexpand\csname atfi@SETGRAYCOLOR##1\noexpand\endcsname
475 }%
476 \else
477 \ifx\HyColor@model\HyColor@model@cmyk
478 \HyColor@DetectPdfVersion

```

```

479 \ifnum\HyColor@PdfVersion<7 %
480 \HyColor@IfModel{#1}{%
481 \HyColor@IfXcolor{%
482 \convertcolorspec\HyColor@model\HyColor@values
483 \HyColor@model@rgb#4%
484 \expandafter\HyColor@NormalizeCommaRGB#4\@nil#4%
485 \edef#4{/C[#4]}% hash-ok
486 }{%
487 \let#4\@empty
488 \HyColor@ErrorModelNoXcolor{#5}{#6}%
489 }%
490 }{%
491 \HyColor@IfXcolor{%
492 \extractcolorspec{#1}#4%
493 \expandafter\convertcolorspec#4%
494 \HyColor@model@rgb#4%
495 \expandafter\HyColor@NormalizeCommaRGB#4\@nil#4%
496 \edef#4{/C[#4]}% hash-ok
497 }{%
498 \let#4\@empty
499 \HyColor@ErrorSpecNoXcolor{#5}{#6}%
500 }%
501 }%
502 \else
503 \edef#4{/C[#3]}% hash-ok
504 \fi
505 \edef#3##1{%
506 #3 %
507 \noexpand\csname atfi@SETCMYKCOLOR##1\noexpand\endcsname
508 }%
509 \else
510 \ifx\HyColor@model\HyColor@model@empty
511 \PackageError{#5}{%
512 Color model `empty' is not permitted for option `#6'%
513 }\@ehc
514 \let#2\@empty
515 \let#3\@gobble
516 \let#4\@empty
517 \else
518 \ifx\HyColor@model\relax % (missing xcolor)
519 \let#3\@gobble
520 \let#4\@empty
521 \else
522 \PackageError{#5}{%
523 Internal error: unsupported color model%
524 }\@ehc
525 \fi
526 \fi
527 \fi
528 \fi
529 \fi
530 \fi
531 }
532 \end{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.

```

533 \*xcolor>
534 \NeedsTeXFormat{LaTeX2e}
535 \ProvidesPackage{xcolor-patch}[2016/05/16 xcolor patch]

536 \RequirePackage{hopatch}[2016/05/16]
537 \hopatch@AfterPackage{xcolor}{%

\XC@ifxcase

538 \long\def\reserved@a#1#2#3{%
539 \long\def\@@tmp##1##2{%
540 \ifx##1#1%
541 \toks@{##2}%
542 \expandafter\remove@to@nnil
543 \else
544 \expandafter\@@tmp
545 \fi
546 }%
547 \@@tmp#2#1{#3}\@nnil\the\toks@
548 }%
549 \ifx\XC@ifxcase\reserved@a
550 \long\def\XC@ifxcase#1#2#3{%
551 \long\def\XC@if##1##2{%
552 \ifx##1#1%
553 \toks@{##2}%
554 \expandafter\remove@to@nnil
555 \else
556 \expandafter\XC@if@
557 \fi
558 }%
559 \XC@if@#2#1{#3}\@nnil
560 \the\toks@
561 }%
562 \fi

\XC@ifcase

563 \long\def\reserved@a#1#2#3{%
564 \long\def\@@tmp##1##2{%
565 \@@expandtwoargs\in@{,##1,}{,##1,}%
566 \ifin@
567 \toks@{##2}%
568 \expandafter\remove@to@nnil
569 \else
570 \expandafter\@@tmp
571 \fi
572 }%
573 \@@tmp#2{#1}{#3}\@nnil
574 \the\toks@
575 }%
```

```

576 \ifx\XC@ifcase\reserved@a
577 \long\def\XC@ifcase#1#2#3{%
578 \long\def\XC@if@##1##2{%
579 \expandtwoargs\in@{, #1, }{, ##1,}%
580 \ifin@
581 \toks@{##2}%
582 \expandafter\remove@to@nnil
583 \else
584 \expandafter\XC@if@
585 \fi
586 }%
587 \XC@if@#2{#1}{#3}\@nnil
588 \the\toks@
589 }%
590 \fi

```

\XC@cnv@gray

```

591 \def\reserved@a#1,{%
592 \XC@ifxcase\tm{%
593 \XC@mod@rgb{%
594 \XC@calcN{#1}\@tmp
595 \edef\@tmp{\@tmp,\@tmp,\@tmp}%
596 }%
597 \XC@mod@cmy{%
598 \XC@calcC{#1}\@tmp
599 \edef\@tmp{\@tmp,\@tmp,\@tmp}%
600 }%
601 \XC@mod@cmyk{%
602 \XC@calcC{#1}\@tmp
603 \edef\@tmp{0,0,0,\@tmp}%
604 }%
605 \XC@mod@RGB{%
606 \edef\@scl{\rangeRGB}%
607 \XC@calcM{#1}\@tmp
608 \edef\@tmp{\@tmp,\@tmp,\@tmp}%
609 }%
610 \XC@mod@HTML{%
611 \edef\@scl{\@cclv}%
612 \XC@calcM{#1}\@tmp
613 \XC@calcH\@tmp\@tmp
614 \edef\@tmp{\@tmp\@tmp\@tmp}%
615 }%
616 \XC@mod@HSB{%
617 \edef\@scl{\rangeHSB}%
618 \XC@calcM{#1}\@tmp
619 \edef\@tmp{0,0,\@tmp}%
620 }%
621 \XC@mod@Gray{%
622 \edef\@scl{\rangeGray}%
623 \XC@calcM{#1}\@tmp
624 }%
625 }%
626 {%
627 \XC@calcN{#1}\@tmp
628 \edef\@tmp{0,0,\@tmp}%
629 }%
630 }%
631 \ifx\XC@cnv@gray\reserved@a

```

```

632 \def\XC@cnv@gray#1,{%
633 \XC@ifxcase\tm{%
634 \XC@mod@rgb{%
635 \XC@calcN{#1}\@tmp
636 \edef\@tmp{\@tmp,\@tmp,\@tmp}%
637 }%
638 \XC@mod@gray{%
639 \XC@mod@cmy{%
640 \XC@calcC{#1}\@tmp
641 \edef\@tmp{\@tmp,\@tmp,\@tmp}%
642 }%
643 \XC@mod@cmyk{%
644 \XC@calcC{#1}\@tmp
645 \edef\@tmp{0,0,0,\@tmp}%
646 }%
647 \XC@mod@RGB{%
648 \edef\@scl{\rangeRGB}%
649 \XC@calcM{#1}\@tmp
650 \edef\@tmp{\@tmp,\@tmp,\@tmp}%
651 }%
652 \XC@mod@HTML{%
653 \edef\@scl{\@cc1v}%
654 \XC@calcM{#1}\@tmp
655 \XC@calcH\@tmp\@tmp
656 \edef\@tmp{\@tmp\@tmp\@tmp}%
657 }%
658 \XC@mod@HSB{%
659 \edef\@scl{\rangeHSB}%
660 \XC@calcM{#1}\@tmp
661 \edef\@tmp{0,0,\@tmp}%
662 }%
663 \XC@mod@Gray{%
664 \edef\@scl{\rangeGray}%
665 \XC@calcM{#1}\@tmp
666 }%
667 }%
668 {%
669 \XC@calcN{#1}\@tmp
670 \edef\@tmp{0,0,\@tmp}%
671 }%
672 }%
673 \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

674 \ifundefined{XC@frameb@x }{%
675 \expandafter\let\csname XC@frameb@x \endcsname\XC@frameb@x
676 \edef\XC@frameb@x{%
677 \noexpand\protect
678 \expandafter\noexpand\csname XC@frameb@x \endcsname

```

```

679 }%
680 \expandafter\ifx\csname XC@frameb@x \endcsname\@frameb@x
681 \let\@frameb@x\XC@frameb@x
682 \fi
683 }{}%
684 }
685 \xcolor

```

### 3 Test

```

686 (*test1)
687 \ProvidesFile{hycolor-test1.tex}[2016/05/16 test file 1]
688 /test1)

689 (*test2)
690 \ProvidesFile{hycolor-test2.tex}[2016/05/16 test file 2]
691 \let\pdfmatch\relax
692 /test2)

693 (test3)\ProvidesFile{hycolor-test3.tex}[2016/05/16 test file 3]
694 (*test)

695 \documentclass{article}
696
697 \usepackage{qstest}
698 \IncludeTests{*}
699 \LogTests{log}{*}{*}
700
701 \makeatletter
702
703 \newcommand*\TestPackageName{test-package}
704 \newcommand*\TestOptionName{test-option}
705
706 \newcommand\Message{}
707 \def\Message#1#{\immediate\write16}
708
709 \newcommand*\ExpectError}[2]{%
710 \begingroup
711 \global\let\saved@errhelp\errhelp
712 \global\let\saved@errmessage\errmessage
713 \let\errhelp\@gobble
714 \def\errmessage##1{%
715 \xdef\@ExpectErrorMessage{##1}%
716 }%
717 \PackageError\TestPackageName{#1}\@ehc
718 \def\errhelp##1{%
719 \global\let\errhelp\saved@errhelp
720 }%
721 \global\let\@ResultErrorMessage\@empty
722 \def\errmessage##1{%
723 \xdef\@ResultErrorMessage{##1}%
724 \global\let\errmessage\saved@errmessage
725 % \Message{[ ##1]}%
726 % \Message{} (ignored error)}%
727 % \Message{}%
728 }%
729 #2%
730 \endgroup

```

```

731 \Expect*{\@ResultErrorMessage}*{\@ExpectErrorMessage}%
732 }
733 \usepackage{scrfile}
734 \usepackage{hycolor}[2016/05/16]
735 /test)
736 (*test1)
737 \begin{qstest}{NumNormalize}{num, normalize}
738 \def\test#1#2{%
739 \HyColor@NormalizeNum{#1}\cmd
740 \Expect*{\cmd}{#2}%
741 }%
742 \test{0}{0}%
743 \test{000}{0}%
744 \test{-1}{0}%
745 \test{ 0 }{0}%
746 \test{1.1}{1}%
747 \test{100}{1}%
748 \test{00100}{1}%
749 \test{99.99}{1}%
750 \test{0.0}{0}%
751 \test{00.00}{0}%
752 \test{0.}{0}%
753 \test{.0}{0}%
754 \test{0.1}{.1}%
755 \test{0.10}{.1}%
756 \test{0.1000}{.1}%
757 \test{.1000}{.1}%
758 \test{0.01}{.01}%
759 \test{0.01010}{.0101}%
760 \test{.0000000001}{.0000000001}%
761 \test{.9999999999}{.9999999999}%
762 \end{qstest}
763
764 \begin{qstest}{BookmarkColor without xcolor}{bookmark, noxcolor}
765 \def\test#1#2{%
766 \HyColor@BookmarkColor{#1}\cmd\TestPackageName\TestOptionName
767 \Expect*{\cmd}{#2}%
768 }%
769 \test{[rgb]{1,0,0}}{1 0 0}%
770 \test{[gray]{0.10}}{.1 .1 .1}%
771 \test{}{}%
772 \test{[rgb]{ 1 , 1 , 0 }}{1 1 0}%
773 \def\errortest[#1]#2{%
774 \ExpectError{%
775 Color model `#1' is not supported\MessageBreak
776 without package `xcolor' in\MessageBreak
777 `TestOptionName=[#1]{#2}'% hash-ok
778 }{%
779 \test{[#1]{#2}}{}% hash-ok
780 }%
781 }%
782 \errortest[cmyk]{1,0,0,0}%
783 \errortest[empty]{}%
784 \def\errortest#1{%
785 \ExpectError{%
786 This color specification is not supported\MessageBreak
787 without package `xcolor' in\MessageBreak
788 `TestOptionName=#1'%

```

```

789     }{%
790         \test{#1}{}%
791     }%
792 }%
793 \end{qstest}
794 /test1}

795 (*test1jtest2)
796 \begin{qstest}{X0134 without xcolor}{X0134, noxcolor}
797     \def\test#1#2{%
798         \HyColor@XZeroOneThreeFour{#1}\cmd\TestPackageName\TestOptionName
799         \Expect*{\cmd}{#2}%
800     }%
801     \test{[empty]{}{}}{%
802         \test{[rgb]{1,0,0}}{1 0 0}%
803         \test{[gray]{0.10}}{.1}%
804         \test{[cmyk]{0,1,0,0}}{0 1 0 0}%
805         \test{[rgb]{ 1 , 1 , 0 }}{1 1 0}%
806     \def\errortest[#1]#2{%
807         \ExpectError{%
808             Color model `#1' is not supported\MessageBreak
809             without package `xcolor' in\MessageBreak
810             `test-option=[#1]{#2}'% hash-ok
811         }{%
812             \HyColor@XZeroOneThreeFour{[#1]}{#2}\cmd
813             \TestPackageName\TestOptionName
814             \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
815         }%
816     }%
817     \errortest[Gray]{10}%
818     \errortest[cmy]{1,0,0}%
819     \def\errortest#1{%
820         \ExpectError{%
821             This color specification is not supported\MessageBreak
822             without package `xcolor' in\MessageBreak
823             `test-option=#1'%
824         }{%
825             \HyColor@XZeroOneThreeFour{#1}\cmd\TestPackageName\TestOptionName
826             \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
827         }%
828     }%
829     \errortest{yellow}%
830 \end{qstest}
831
832 \begin{qstest}{HyperrefBorderColor without xcolor}%
833     {hyperref bordercolor, noxcolor}%
834     \def\test#1#2{%
835         \HyColor@HyperrefBorderColor{#1}\cmd\TestPackageName\TestOptionName
836         \Expect*{\cmd}{#2}%
837     }%
838     \test{[rgb]{1,0,0}}{1 0 0}%
839     \test{[gray]{0.10}}{.1 .1 .1}%
840     \test{[rgb]{ 1 , 1 , 0 }}{1 1 0}%
841     \def\errortest[#1]#2{%
842         \ExpectError{%
843             Color model `#1' is not supported\MessageBreak
844             without package `xcolor' in\MessageBreak
845             `test-option=[#1]{#2}'% hash-ok
846         }{%

```



```

847 \HyColor@HyperrefBorderColor{[#1]{#2}}\cmd
848 \TestPackageName\TestOptionName
849 \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
850 }%
851 }%
852 \errortest[Gray]{10}%
853 \errortest[cmy]{1,0,0}%
854 \errortest[cmk]{0,1,0,0}%
855 \def\errortest#1{%
856 \ExpectError{%
857 This color specification is not supported\MessageBreak
858 without package `xcolor' in\MessageBreak
859 `test-option=#1'%
860 }{%
861 \HyColor@HyperrefBorderColor{#1}\cmd
862 \TestPackageName\TestOptionName
863 \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
864 }%
865 }%
866 \errortest{yellow}%
867 \end{qstest}
868 /test1 j test2)
869 (*test1 j test2)
870 \usepackage{xcolor}
871 \definecolor[named]{MyGreen}{rgb}{0,0.7,0}
872 \definecolor{mygreen}{named}{MyGreen}
873 /test1 j test2)
874 (*test1)
875 \begin{qstest}{BookmarkColor with xcolor}{bookmark, xcolor}
876 \def\test#1#2{%
877 \HyColor@BookmarkColor{#1}\cmd\PackageName\OptionName
878 \Expect*{\cmd}{#2}%
879 }%
880 \test{[rgb]{1,0,0}}{1 0 0}%
881 \test{[gray]{0.10}}{.1 .1 .1}%
882 \test{}{}%
883 \test{[rgb]{ 1 , 1 , 0 }}{1 1 0}%
884 \test{[cmyk]{1,0,0,0}}{0 1 1}%
885 \test{red}{1 0 0}%
886 \test{cyan}{0 1 1}%
887 \test{red!40!blue}{.4 0 .6}%
888 \test{[Gray]{10}}{.66667 .66667 .66667}%
889 \test{[RGB]{100,200,50}}{.39217 .78432 .19609}%
890 \test{[wave]{363}}{.00316 0 .00316}%
891 \test{[wave]814}{.00797 0 0}%
892 \test{[HSB]{100,200,50}}{.03473 .20833 .12152}%
893 \test{[HTML]{A800FF}}{.65881 0 1}%
894 \test{[cmy]{.3,.5,.2}}{.7 .5 .8}%
895 \test{[cmyk]{.3,.5,.2,.1}}{.6 .4 .7}%
896 \test{[hsb]{.3,.5,.2}}{.12 .2 .1}%
897 \test{[Hsb]{120,.5,.2}}{.1 .2 .1}%
898 \test{[tHsb]{120,.5,.2}}{.2 .2 .1}%
899 \test{[named]{MyGreen}}{0 .7 0}%
900 \test{mygreen}{0 .7 0}%
901 \end{qstest}
902
903 \begin{qstest}{HyperrefColor}{hyperref, color}
904 \def\test#1#2{%

```

```

905 \HyColor@HyperrefColor{#1}\cmd
906 \Expect*{\cmd}{#2}%
907 }%
908 \test{red}{red}%
909 \test{[rgb]{1,0,0}}{[rgb]{1,0,0}}%
910 \HyColor@HyperrefColor{}\cmd
911 \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
912 \end{qstest}
913 \test1}
914 (*test1 j test2)
915 \begin{qstest}{X0134 with xcolor}{hyperref, X0134, xcolor}
916 \def\test#1#2{%
917 \HyColor@XZeroOneThreeFour{#1}\cmd\PackageName\OptionName
918 \Expect*{\cmd}{#2}%
919 }%
920 \test{[empty]{}{}}%
921 \test{[gray]{0.1}}{.1}%
922 \test{[rgb]{1,0.5,0.0}}{1 .5 0}%
923 \test{[cmyk]{0,1,0,0.5}}{0 1 0 .5}%
924 \test{[Gray]{10}}{.66667}%
925 \test{red}{1 0 0}%
926 \test{1 0 0}{1 0 0}%
927 \test{001.0 .23 0}{1 .23 0}%
928 \test{[named]{MyGreen}}{0 .7 0}%
929 \test{mygreen}{0 .7 0}%
930 \HyColor@XZeroOneThreeFour{}\cmd\PackageName\OptionName
931 \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
932 \end{qstest}
933
934 \begin{qstest}{FieldColor}{hyperref, field, FieldColor}
935 \def\test#1#2{%
936 \HyColor@FieldColor{#1}\cmd\PackageName\OptionName
937 \Expect*{\cmd}{#2}%
938 }%
939 \test{}{}%
940 \test{[gray]{0.7}}{.7 g}%
941 \test{[rgb]{1,0,0}}{1 0 0 rg}%
942 \test{[cmyk]{0,1,0,0}}{0 1 0 0 k}%
943 \test{[cmy]{.5,.4,.3}}{.5 .6 .7 rg}%
944 \end{qstest}
945 \test1 j test2}

```

### 3.1 Test for package attachfile2

```

946 (*test3)
947 \def\atfi@SETRGBCOLORtest{set-rgb}
948 \def\atfi@SETGRAYCOLORtest{set-gray}
949 \def\atfi@SETCMYKCOLORtest{set-cmyk}
950 \def\Test#1#2#3#4#5{%
951 \begingroup
952 \setbox0=\hbox{%
953 \begingroup
954 \chardef\HyColor@PdfVersion=6 %
955 \HyColor@AttachfileColor{#1}\spec\inlinemacro\annot
956 \TestPackageName\TestOptionName
957 \edef\inline{\inlinemacro{test}}%
958 \expandafter\Expect\expandafter{\spec}{#2}%
959 \expandafter\Expect\expandafter{\inline}{#3}%
960 \expandafter\Expect\expandafter{\annot}{#4}%

```

```

961 \endgroup
962 \begingroup
963 \chardef\HyColor@PdfVersion=7 %
964 \HyColor@AttachfileColor{#1}\spec\inlinemacro\annot
965 \TestPackageName\TestOptionName
966 \edef\inline{\inlinemacro{test}}%
967 \expandafter\Expect\expandafter{\spec}{#2}%
968 \expandafter\Expect\expandafter{\inline}{#3}%
969 \expandafter\Expect\expandafter{\annot}{#5}%
970 \endgroup
971 }%
972 \Expect*{\the\wd0}{0.0pt}%
973 \endgroup
974 }
975 \newif\ifError
976 \def\TestError[#1]#2#3#4#5#6{%
977 \begingroup
978 \global\Errorfalse
979 \let\OrgPackageError\PackageError
980 \def\PackageError##1##2##3{%
981 \edef\TestTemp{##1}%
982 \ifx\TestTemp\TestPackageName
983 \Expect*{\ifError too many errors\else ok\fi}{ok}%
984 \Expect*{#6}*{##2}%
985 \global\Errortrue
986 \else
987 \OrgPackageError{##1}{##2}{##3}%
988 \fi
989 }%
990 \setbox0=\hbox{%
991 \begingroup
992 \chardef\HyColor@PdfVersion=#1 %
993 \HyColor@AttachfileColor{#2}\spec\inlinemacro\annot
994 \TestPackageName\TestOptionName
995 \edef\inline{\inlinemacro{test}}%
996 \expandafter\Expect\expandafter{\spec}{#3}%
997 \expandafter\Expect\expandafter{\inline}{#4}%
998 \expandafter\Expect\expandafter{\annot}{#5}%
999 \endgroup
1000 \ifx\#6\%
1001 \else
1002 \Expect*{\ifError ok\else missing error\fi}{ok}%
1003 \fi
1004 }%
1005 \Expect*{\the\wd0}{0.0pt}%
1006 \endgroup
1007 }
1008 \def\NoEmptyModel{%
1009 Color model `empty' is not permitted for option ``TestOptionName'%
1010 }
1011 \def\ModelNoXcolor#1#2{%
1012 Color model `#1' is not supported\MessageBreak
1013 without package `xcolor' in\MessageBreak
1014 ``TestOptionName=[#1]{#2}'% hash-ok
1015 }
1016 \def\SpecNoXColor#1{%
1017 This color specification is not supported\MessageBreak
1018 without package `xcolor' in\MessageBreak

```

```

1019 `test-option=#1'%
1020 }

1021 \begin{qstest}{AttachfileColor}{AttachfileColor}
1022   \Test{}{}{}{}{}%
1023   \Test{0.1 0.2 0.3}{{rgb}{{.1,.2,.3}}{.1 .2 .3 set-rgb}%
1024     {/C[.1 .2 .3]}{/C[.1 .2 .3]}}%
1025   \Test{{gray}{0.4}}{{gray}{0.4}}{.4 set-gray}%
1026     {/C[.4 .4 .4]}{/C[.4]}%
1027   \Test{{rgb}{0.3,.2,.1}}{{rgb}{0.3,.2,.1}}{.3 .2 .1 set-rgb}%
1028     {/C[.3 .2 .1]}{/C[.3 .2 .1]}}%
1029   \Test{0.0 1.0 1}{{rgb}{0,1,1}}{0 1 1 set-rgb}%
1030     {/C[0 1 1]}{/C[0 1 1]}}%
1031   \Test{{gray}1}{{gray}1}{1 set-gray}{/C[1 1 1]}{/C[1]}%
1032   \TestError[6]{{empty}{}{}{}{}{}{}\NoEmptyModel
1033   \TestError[7]{{empty}{}{}{}{}{}{}\NoEmptyModel
1034   \TestError[6]{{cmyk}{{.1,.2,.3,.4}}{cmyk}{{.1,.2,.3,.4}}}%
1035     {.1 .2 .3 .4 set-cmyk}{}%
1036     {\ModelNoXcolor{cmyk}{{.1,.2,.3,.4}}}%
1037   \TestError[7]{{cmyk}{{.1,.2,.3,.4}}{cmyk}{{.1,.2,.3,.4}}}%
1038     {.1 .2 .3 .4 set-cmyk}{/C[.1 .2 .3 .4]}{}%
1039   \TestError[6]{red}{red}{}{}{}\SpecNoXColor{red}}%
1040   \TestError[7]{red}{red}{}{}{}\SpecNoXColor{red}}%
1041 \end{qstest}

1042 \usepackage{xcolor}
1043 \definecolor[named]{MyGreen}{rgb}{0,0.7,0}
1044 \definecolor{mygreen}{named}{MyGreen}
1045 \definecolor{graynine}{gray}{0.9}
1046 \definecolor{GraySix}{Gray}{9}

1047 \begin{qstest}{AttachfileColorX}{AttachfileColorX}
1048   \Test{}{}{}{}{}%
1049   \Test{0.1 0.2 0.3}{{rgb}{{.1,.2,.3}}{.1 .2 .3 set-rgb}%
1050     {/C[.1 .2 .3]}{/C[.1 .2 .3]}}%
1051   \Test{{gray}{0.4}}{{gray}{0.4}}{.4 set-gray}%
1052     {/C[.4 .4 .4]}{/C[.4]}%
1053   \Test{{rgb}{0.3,.2,.1}}{{rgb}{0.3,.2,.1}}{.3 .2 .1 set-rgb}%
1054     {/C[.3 .2 .1]}{/C[.3 .2 .1]}}%
1055   \Test{0.0 1.0 1}{{rgb}{0,1,1}}{0 1 1 set-rgb}%
1056     {/C[0 1 1]}{/C[0 1 1]}}%
1057   \Test{{gray}1}{{gray}1}{1 set-gray}{/C[1 1 1]}{/C[1]}%
1058   \Test{red}{red}{1 0 0 set-rgb}{/C[1 0 0]}{/C[1 0 0]}}%
1059   \Test{black}{black}{0 set-gray}{/C[0 0 0]}{/C[0]}%
1060   \Test{cyan}{cyan}{1 0 0 0 set-cmyk}{/C[0 1 1]}{/C[1 0 0 0]}}%
1061   \Test{{named}{black}}{[named]{black}}{0 0 0 set-rgb}%
1062     {/C[0 0 0]}{/C[0 0 0]}}%
1063   \Test{{Gray}{9}}{[Gray]{9}}{.6 set-gray}{/C[.6 .6 .6]}{/C[.6]}}%
1064   \Test{{HTML}{0080FF}}{[HTML]{0080FF}}{0 .50195 1 set-rgb}%
1065     {/C[0 .50195 1]}{/C[0 .50195 1]}}%
1066   \Test{graynine}{graynine}{.9 set-gray}{/C[.9 .9 .9]}{/C[.9]}}%
1067   \Test{GraySix}{GraySix}{.6 set-gray}{/C[.6 .6 .6]}{/C[.6]}}%
1068   \Test{{wave}{500}}{[wave]{500}}{0 1 .49846 set-rgb}%
1069     {/C[0 1 .49846]}{/C[0 1 .49846]}}%
1070   \TestError[6]{{empty}{}{}{}{}{}{}\NoEmptyModel
1071   \TestError[7]{{empty}{}{}{}{}{}{}\NoEmptyModel
1072 \end{qstest}

1073 \end{test3}

1074 \end{test}

```

```

1075 \begin{document}
1076 \end{document}
1077 \test)

```

### 3.2 Test for package xcolor

```

1078 <*test-xcolor>
1079 \NeedsTeXFormat{LaTeX2e}
1080 \nofiles
1081 \documentclass{minimal}
1082 <*xcol1>
1083 \usepackage{xcolor}
1084 \usepackage{xcolor-patch}[2016/05/16]
1085 </xcol1>
1086 <*xcol2>
1087 \usepackage{scrfile}
1088 \usepackage{xcolor-patch}[2016/05/16]
1089 \usepackage{xcolor}
1090 </xcol2>
1091 <*xcol3>
1092 \usepackage{xcolor-patch}[2016/05/16]
1093 \usepackage{xcolor}
1094 \begin{document}
1095 </xcol3>
1096 \makeatletter
1097 \newcommand*{\ColModList}{%
1098   rgb,%
1099   cmy,%
1100   cmyk,%
1101   hsb,%
1102   Hsb,%
1103   tHsb,%
1104   gray,%
1105   RGB,%
1106   HTML,%
1107   HSB,%
1108   Gray,%
1109   % wave,
1110 }
1111 \newcommand*{\StartModel}{rgb}
1112 \newcommand*{\StartValues}{.1,.2,.3}
1113 \@for\x:=\ColModList\do{%
1114   \ifx\x\@empty
1115   \else
1116     \convertcolorspec\StartModel\StartValues\x\y
1117     \typeout{* [\StartModel]{\StartValues} ==> [\x]{\y}}%
1118     \@for\xx:=\ColModList\do{%
1119       \ifx\xx\@empty
1120       \else
1121         \convertcolorspec\x\y\xx\yy
1122         \typeout{* [\x]{\y} ==> [\xx]{\yy}}%
1123       \fi
1124     }%
1125   \fi
1126 }
1127 <xcol3>\end{document}
1128 <xcol1jxcol2>\@@end
1129 </test-xcolor>

```

### 3.2.1 Test for \@frameb@x/\fbox

```
1130 \test-xcolor-fbox>
1131 \NeedsTeXFormat{LaTeX2e}
1132 \documentclass{article}
1133 \usepackage{xcolor}
1134 \usepackage{xcolor-patch}[2016/05/16]
1135 \makeatletter
1136 \protected@edef\x{\fbox{abc}}
1137 \let\@tempa\@undefined
1138 \protected@edef\x{\fbox{abc}}
1139 \makeatother
1140 \begin{document}
1141 \MakeUppercase{\fbox{abc}}
1142 \end{document}
1143 \test-xcolor-fbox>
```

## 4 Installation

### 4.1 Download

**Package.** This package is available on CTAN<sup>1</sup>:

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

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

**Bundle.** All the packages of the bundle ‘oberdiek’ 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/oberdiek.tds.zip](#)

*TDS* refers to the standard “A Directory Structure for T<sub>E</sub>X Files” ([CTAN:tds/tds.pdf](#)). Directories with `texmf` in their name are usually organized this way.

### 4.2 Bundle installation

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

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

**Script installation.** Check the directory `TDS:scripts/oberdiek/` for scripts that need further installation steps.

### 4.3 Package installation

**Unpacking.** The `.dtx` file is a self-extracting docstrip archive. The files are extracted by running the `.dtx` through plain T<sub>E</sub>X:

```
tex hycolor.dtx
```

---

<sup>1</sup>[CTAN:pkg/hycolor](#)

**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/oberdiek/hycolor.sty
xcolor-patch.sty      → tex/latex/oberdiek/xcolor-patch.sty
hycolor.pdf           → doc/latex/oberdiek/hycolor.pdf
test/hycolor-test1.tex → doc/latex/oberdiek/test/hycolor-test1.tex
test/hycolor-test2.tex → doc/latex/oberdiek/test/hycolor-test2.tex
test/hycolor-test3.tex → doc/latex/oberdiek/test/hycolor-test3.tex
test/hycolor-test-xcol1.tex → doc/latex/oberdiek/test/hycolor-test-xcol1.tex
test/hycolor-test-xcol2.tex → doc/latex/oberdiek/test/hycolor-test-xcol2.tex
test/hycolor-test-xcol3.tex → doc/latex/oberdiek/test/hycolor-test-xcol3.tex
test/hycolor-test-xcol4.tex → doc/latex/oberdiek/test/hycolor-test-xcol4.tex
hycolor.dtx           → source/latex/oberdiek/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`.

## 4.4 Refresh file name databases

If your  $\text{\TeX}$  distribution (`te $\text{\TeX}$` , `mik $\text{\TeX}$` , ...) relies on file name databases, you must refresh these. For example, `te $\text{\TeX}$`  users run `texhash` or `mktextlsr`.

## 4.5 Some details for the interested

**Unpacking with  $\text{\LaTeX}$ .** The `.dtx` chooses its action depending on the format:

**plain  $\text{\TeX}$ :** Run `docstrip` and extract the files.

**$\text{\LaTeX}$ :** Generate the documentation.

If you insist on using  $\text{\LaTeX}$  for `docstrip` (really, `docstrip` does not need  $\text{\LaTeX}$ ), 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.

**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 `pdf $\text{\LaTeX}$` :

```

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

```

# 5 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.

## 6 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		608, 612, 613, 614, 618, 619,
<code>\!</code>	7	623, 627, 628, 635, 636, 640,
<code>\"</code>	12	641, 644, 645, 649, 650, 654,
<code>\+</code>	10	655, 656, 660, 661, 665, 669, 670
<code>\-</code>	9	<code>\@ExpectErrorMessage</code> . . . . . 715, 731
<code>\.</code>	365	<code>\@ResultErrorMessage</code> . . 721, 723, 731
<code>\:</code>	8	<code>\@ReturnAfterFi</code> . . . . . 65, 70, <u>163</u>
<code>\;</code>	11	<code>\@car</code> . . . . . 350
<code>\&gt;</code>	13	<code>\@ccclv</code> . . . . . 611, 653
<code>\@end</code>	1128	<code>\@ehc</code> . . . . . 136, 143, 299, 513, 524, 717
<code>\@scl</code>	606,	<code>\@empty</code> . . . 40, 50, 51, 111, 118, 125,
	611, 617, 622, 648, 653, 659, 664	185, 205, 213, 244, 285, 288,
<code>\@tmp</code>	539,	290, 347, 406, 447, 449, 487,
	544, 547, 564, 570, 573, 594,	498, 514, 516, 520, 721, 1114, 1119
	595, 598, 599, 602, 603, 607,	<code>\@expandtwoargs</code> . . . . . 565, 579





294, 296, 335, 370, 387, 391, 394, 458, 465, 477, 482, 510, 518	\ifx ..... 46, 51, 62, 72, 103, 106, 118, 166, 183, 185, 205, 212, 214, 217, 219, 221, 244, 254, 257, 260, 263, 287, 290, 292, 294, 296, 306, 323, 347, 351, 353, 355, 391, 394, 406, 426, 435, 447, 458, 465, 477, 510, 518, 540, 549, 552, 576, 631, 680, 814, 826, 849, 863, 911, 931, 982, 1000, 1114, 1119
\HyColor@model@cmyk ..... ..... 178, 179, 219, 260, 296, 477	\immediate ..... 707
\HyColor@model@empty 172, 173, 212, 510	\in@ ..... 565, 579
\HyColor@model@Gray 180, 181, 221, 263	\IncludeTests ..... 698
\HyColor@model@gray ..... ..... 106, 174, 175, 214, 224, 226, 254, 265, 268, 292, 394, 465	\inline ... 957, 959, 966, 968, 995, 997
\HyColor@model@rgb ..... 100, 103, 122, 176, 177, 217, 234, 236, 257, 271, 273, 294, 335, 370, 388, 391, 414, 458, 483, 494	\inlinemacro 955, 957, 964, 966, 993, 995
\HyColor@NormalizeCommaCMYK .... ..... 86, 220, 261	
\HyColor@NormalizeCommaRGB ..... ..... 78, 101, 104, 123, 218, 235, 248, 258, 272, 389, 392, 410, 415, 484, 495	<b>L</b>
\HyColor@NormalizeNum ..... .... 36, 79, 81, 83, 87, 89, 91, 93, 107, 215, 225, 255, 266, 395, 739	\LogTests ..... 699
\HyColor@one ..... 343, 353	<b>M</b>
\HyColor@PdfVersion ..... . 427, 429, 467, 479, 954, 963, 992	\makeatletter ..... 701, 1096, 1135
\HyColor@resultfalse ..... ..... 332, 348, 357, 379, 451	\makeatother ..... 1139
\HyColor@resulttrue ..... 321, 376	\MakeUppercase ..... 1141
\HyColor@ReverseString . 49, 54, 61, 66	\Message ..... 706, 707, 725, 726, 727
\HyColor@SpaceToComma ..... 433, 455	\MessageBreak ..... 133, 134, 140, 141, 775, 776, 786, 787, 808, 809, 821, 822, 843, 844, 857, 858, 1012, 1013, 1017, 1018
\HyColor@StripLeadingZeros 50, 71, 73	\ModelNoXcolor ..... 1011, 1036
\HyColor@temp ..... .... 79, 80, 81, 82, 83, 84, 87, 88, 89, 90, 91, 92, 93, 94, 322, 323, 346, 347, 350, 351, 353, 355	<b>N</b>
\HyColor@TwoSpaces ..... 319, 323	\NeedsTeXFormat ... 2, 534, 1079, 1131
\HyColor@UseColor ..... 182	\newcommand ..... 703, 704, 706, 709, 1097, 1111, 1112
\HyColor@values . 99, 104, 108, 117, 135, 142, 155, 160, 202, 204, 216, 218, 220, 223, 233, 243, 247, 248, 251, 253, 256, 259, 262, 264, 267, 270, 336, 371, 387, 392, 396, 405, 409, 410, 482	\newif ..... 304, 975
\HyColor@WithModel ..... 147, 153	\NoEmptyModel ..... .... 1008, 1032, 1033, 1070, 1071
\HyColor@WithoutModel ..... 149, 158	\nofiles ..... 1080
\HyColor@XZeroOneThreeFour ..... ..... 210, 283, 286, 452, 798, 812, 825, 917, 930	<b>O</b>
\HyColor@zero ..... 342, 351	\OptionName ..... 877, 917, 930, 936
	\OrgPackageError ..... 979, 987
<b>I</b>	<b>P</b>
\ifdim ..... 37	\PackageError ..... 132, 139, 299, 511, 522, 717, 979, 980
\ifError ..... 975, 983, 1002	\PackageName ..... 877, 917, 930, 936
\ifHyColor@result ..... ..... 304, 325, 327, 334, 453	\pdflastmatch ..... 372, 373, 374
\ifin@ ..... 566, 580	\pdfmatch ..... 368, 691
\ifnum ..... 45, 368, 467, 479	\protect ..... 677
	\protected@edef ..... 1136, 1138
	\ProvidesFile ..... 687, 690, 693
	\ProvidesPackage ..... 3, 535
	<b>R</b>
	\rangeGray ..... 622, 664
	\rangeHSB ..... 617, 659
	\rangeRGB ..... 606, 648
	\remove@to@nnil ... 542, 554, 568, 582

\RequirePackage	5, 536	\tm	592, 633
\reserved@a	538, 549, 563, 576, 591, 631	\toks@	541, 547, 553, 560, 567, 574, 581, 588
<b>S</b>			
\saved@errhelp	711, 719	\typeout	1117, 1122
\saved@errmessage	712, 724	<b>U</b>	
\setbox	952, 990	\usepackage	697, 733, 734, 870, 1042, 1083, 1084, 1087, 1088, 1089, 1092, 1093, 1133, 1134
\space	319, 368, 369	<b>W</b>	
\spec	955, 958, 964, 967, 993, 996	\wd	972, 1005
\SpecNoXColor	1016, 1039, 1040	\write	707
\StartModel	1111, 1116, 1117	<b>X</b>	
\StartValues	1112, 1116, 1117	\x	14, 28, 31, 35, 313, 318, 1113, 1114, 1116, 1117, 1121, 1122, 1136, 1138
\strip@prefix	372, 373, 374	\XC@calcC	598, 602, 640, 644
<b>T</b>			
\Test	950, 1022, 1023, 1025, 1027, 1029, 1031, 1048, 1049, 1051, 1053, 1055, 1057, 1058, 1059, 1060, 1061, 1063, 1064, 1066, 1067, 1068	\XC@calcH	613, 655
\test	738, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 765, 769, 770, 771, 772, 779, 790, 797, 801, 802, 803, 804, 805, 834, 838, 839, 840, 876, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 904, 908, 909, 916, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 935, 939, 940, 941, 942, 943	\XC@calcM	607, 612, 618, 623, 649, 654, 660, 665
\TestError	976, 1032, 1033, 1034, 1037, 1039, 1040, 1070, 1071	\XC@calcN	594, 627, 635, 669
\TestOptionName	704, 766, 777, 788, 798, 813, 825, 835, 848, 862, 956, 965, 994, 1009, 1014	\XC@cnv@gray	591
\TestPackageName	703, 717, 766, 798, 813, 825, 835, 848, 862, 956, 965, 982, 994	\XC@frameb@x	675, 676, 681
\TestTemp	981, 982	\XC@if@	551, 556, 559, 578, 584, 587
\the	547, 560, 574, 588, 972, 1005	\XC@ifcase	563
		\XC@ifxcase	538, 592, 633
		\XC@mod@cmy	597, 639
		\XC@mod@cmyk	601, 643
		\XC@mod@Gray	621, 663
		\XC@mod@gray	638
		\XC@mod@HSB	616, 658
		\XC@mod@HTML	610, 652
		\XC@mod@RGB	605, 647
		\XC@mod@rgb	593, 634
		\xx	1118, 1119, 1121, 1122
<b>Y</b>			
		\y	23, 35, 1116, 1117, 1121, 1122
		\yy	1121, 1122
<b>Z</b>			
		\z@	37, 369
		\zap@space	40