

The hycolor package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2008/08/01 v1.3

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	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 bookmark	5
2.4	Utils	7
2.5	Package hyperref	8
2.5.1	Options Hyp.*color	8
2.5.2	Generic algorithm	9
2.5.3	Field options	11
2.5.4	Detection for naked RGB values	11
2.5.5	Options *bordercolor	13
2.6	Package attachfile2	14
2.7	Patch for package xcolor	16
3	Test	19
3.1	Test for package attachfile2	23
3.2	Test for package xcolor	25
4	Installation	26
4.1	Download	26
4.2	Bundle installation	27
4.3	Package installation	27
4.4	Refresh file name databases	27
4.5	Some details for the interested	27
5	History	28
	[2007/04/09 v1.0]	28
	[2007/04/11 v1.1]	28
	[2008/07/29 v1.2]	28
	[2008/08/01 v1.3]	28
6	Index	28

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.

```
\hyperref{linkbordercolor=red}% needs xcolor
\hyperref{linkbordercolor=[named]{red}% needs xcolor
\hyperref{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 <code>bookmark</code>
Hyp	Package <code>hyperref</code> : package options or <code>\hypersetup</code>
Field	Package <code>hyperref</code> : Form field options
AtFi	Package <code>attachfile2</code> : option <code>color</code>

2 Implementation

```

1 <*package>
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{hycolor}%
4 [2008/08/01 v1.3 Code for color options of hyperref/bookmark (H0)]%
5 \RequirePackage{xcolor-patch}[2008/08/01]

```

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 \leftarrow 0;

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

cmd \leftarrow number after dot of *value*;

cmd \leftarrow strip trailing zeros from *cmd*;

if dot remains only **then**

cmd \leftarrow 0;

end

else

cmd \leftarrow 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
    cmd ← no color;
  end
  case with model
    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
  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
    cmd ← no color;
  end
  case with model
    Call \color with model;
  end
  case without model
    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
    cmd ← no color;
  end
  case with model
    switch model do
      case empty
        cmd ← "";
      end
      case gray, rgb, cmyk
        cmd ← output();
      end
      case Gray
        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
  (model, values) ← ("rgb", (r,g,b));
  cmd ← output();
end
case without model
  if with xcolor then
    (model, values) ← get model and values(value);
    switch model do
      case gray, rgb, cmyk
        cmd ← output();
      end
      case Gray
        (model, values) ← convert to gray;
        cmd ← output();
      end
      else
        (model, values) ← convert to rgb;
        cmd ← output();
      end
    end
  end
  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\csgname pdfmatch\endcsgname\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\@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\pdfmatch2,%
374 \expandafter\strip@prefix\pdfmatch3%
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
    cmd  $\leftarrow$  no color;
  end
  case with model
    if with xcolor then
      (model, values)  $\leftarrow$  convert to rgb;
      cmd  $\leftarrow$  output values;
    else
      switch model do
        case rgb, gray
          cmd  $\leftarrow$  output values;
        end
        else
          error(package, option, "Missing xcolor");
          cmd  $\leftarrow$  no color;
        end
      end
    end
  end
  case rgb values
    cmd  $\leftarrow$  output values;
  end
  case without model
    if with xcolor then
      (model, values)  $\leftarrow$  convert to rgb;
      cmd  $\leftarrow$  output values;
    else
      error(package, option, "Missing xcolor"); cmd  $\leftarrow$  no color;
    end
  end
end
end
end

```

`\HyColor@HyperrefBorderColor`

```

384 \def\HyColor@HyperrefBorderColor#1#2#3#4{%
385   \HyColor@IfModel{#1}{%
386     \HyColor@IfXcolor{%
387       \convertcolspec\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\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 </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}[2008/08/01 xcolor patch]
536 \@ifpackageloaded{xcolor}{%
537     \@firstofone
538 }{%
539     \@ifpackageloaded{scrfile}{%
540         \AfterPackage{xcolor}%
541     }{%
542         \def\reserved@a{%
543             \edef\x{%
544                 \endgroup
545                 \noexpand\AtBeginDocument{%
546                     \noexpand\@ifpackageloaded{xcolor}{\the\toks@}{}%
547                 }%
548             }%
549             \x
550         }%
551         \begingroup
552         \afterassignment\reserved@a
553         \toks@
554     }%

```



```

555 }%
556 {%

\XC@ifxcase

557 \long\def\reserved@a#1#2#3{%
558 \long\def\@tmp##1##2{%
559 \ifx#1##1%
560 \toks@{##2}%
561 \expandafter\remove@to@nnil
562 \else
563 \expandafter\@tmp
564 \fi
565 }%
566 \@tmp#2#1{#3}\@nnil\the\toks@
567 }%
568 \ifx\XC@ifxcase\reserved@a
569 \long\def\XC@ifxcase#1#2#3{%
570 \long\def\XC@if@##1##2{%
571 \ifx#1##1%
572 \toks@{##2}%
573 \expandafter\remove@to@nnil
574 \else
575 \expandafter\XC@if@
576 \fi
577 }%
578 \XC@if@#2#1{#3}\@nnil
579 \the\toks@
580 }%
581 \fi

\XC@ifcase

582 \long\def\reserved@a#1#2#3{%
583 \long\def\@tmp##1##2{%
584 \@expandtwoargs\in@{, #1, }{, ##1, }%
585 \ifin@
586 \toks@{##2}%
587 \expandafter\remove@to@nnil
588 \else
589 \expandafter\@tmp
590 \fi
591 }%
592 \@tmp#2{#1}{#3}\@nnil
593 \the\toks@
594 }%
595 \ifx\XC@ifcase\reserved@a
596 \long\def\XC@ifcase#1#2#3{%
597 \long\def\XC@if@##1##2{%
598 \@expandtwoargs\in@{, #1, }{, ##1, }%
599 \ifin@
600 \toks@{##2}%
601 \expandafter\remove@to@nnil
602 \else
603 \expandafter\XC@if@
604 \fi
605 }%
606 \XC@if@#2{#1}{#3}\@nnil
607 \the\toks@
608 }%
609 \fi

\XC@cvt@gray

610 \def\reserved@a#1,{%

```

```

611 \XC@ifxcase\tm{%
612 \XC@mod@rgb{%
613 \XC@calcN{#1}\@tmp
614 \edef\@tmp{\@tmp,\@tmp,\@tmp}%
615 }%
616 \XC@mod@cmy{%
617 \XC@calcC{#1}\@tmp
618 \edef\@tmp{\@tmp,\@tmp,\@tmp}%
619 }%
620 \XC@mod@cmyk{%
621 \XC@calcC{#1}\@tmp
622 \edef\@tmp{0,0,0,\@tmp}%
623 }%
624 \XC@mod@RGB{%
625 \edef\@scl{\rangeRGB}%
626 \XC@calcM{#1}\@tmp
627 \edef\@tmp{\@tmp,\@tmp,\@tmp}%
628 }%
629 \XC@mod@HTML{%
630 \edef\@scl{\@cclv}%
631 \XC@calcM{#1}\@tmp
632 \XC@calcH\@tmp\@tmp
633 \edef\@tmp{\@tmp\@tmp\@tmp}%
634 }%
635 \XC@mod@HSB{%
636 \edef\@scl{\rangeHSB}%
637 \XC@calcM{#1}\@tmp
638 \edef\@tmp{0,0,\@tmp}%
639 }%
640 \XC@mod@Gray{%
641 \edef\@scl{\rangeGray}%
642 \XC@calcM{#1}\@tmp
643 }%
644 }%
645 {%
646 \XC@calcN{#1}\@tmp
647 \edef\@tmp{0,0,\@tmp}%
648 }%
649 }%
650 \ifx\XC@cnv@gray\reserved@a
651 \def\XC@cnv@gray#1,{%
652 \XC@ifxcase\tm{%
653 \XC@mod@rgb{%
654 \XC@calcN{#1}\@tmp
655 \edef\@tmp{\@tmp,\@tmp,\@tmp}%
656 }%
657 \XC@mod@gray{}}%
658 \XC@mod@cmy{%
659 \XC@calcC{#1}\@tmp
660 \edef\@tmp{\@tmp,\@tmp,\@tmp}%
661 }%
662 \XC@mod@cmyk{%
663 \XC@calcC{#1}\@tmp
664 \edef\@tmp{0,0,0,\@tmp}%
665 }%
666 \XC@mod@RGB{%
667 \edef\@scl{\rangeRGB}%
668 \XC@calcM{#1}\@tmp
669 \edef\@tmp{\@tmp,\@tmp,\@tmp}%
670 }%
671 \XC@mod@HTML{%
672 \edef\@scl{\@cclv}%

```

```

673      \XC@calcM{#1}\@@tmp
674      \XC@calcH\@@tmp\@@tmp
675      \edef\@@tmp{\@@tmp\@@tmp\@@tmp}%
676      }%
677      \XC@mod@HSB{%
678      \edef\@@scl{\rangeHSB}%
679      \XC@calcM{#1}\@@tmp
680      \edef\@@tmp{0,0,\@@tmp}%
681      }%
682      \XC@mod@Gray{%
683      \edef\@@scl{\rangeGray}%
684      \XC@calcM{#1}\@@tmp
685      }%
686      }%
687      {%
688      \XC@calcN{#1}\@@tmp
689      \edef\@@tmp{0,0,\@@tmp}%
690      }%
691      }%
692  \fi
693 }

694 </xcolor>

```

3 Test

```

695 <*test1>
696 \ProvidesFile{hycolor-test1.tex}[2008/08/01 test file 1]
697 </test1>

698 <*test2>
699 \ProvidesFile{hycolor-test2.tex}[2008/08/01 test file 2]
700 \let\pdfmatch\relax
701 </test2>

702 <test3>\ProvidesFile{hycolor-test3.tex}[2008/08/01 test file 3]

703 <*test>

704 \documentclass{article}
705
706 \usepackage{qstest}
707 \IncludeTests{*}
708 \LogTests{log}{*}{*}
709
710 \makeatletter
711
712 \newcommand*{\TestPackageName}{test-package}
713 \newcommand*{\TestOptionName}{test-option}
714
715 \newcommand\Message{}
716 \def\Message#1#{\immediate\write16}
717
718 \newcommand*{\ExpectError}[2]{%
719   \begingroup
720     \global\let\saved@errhelp\errhelp
721     \global\let\saved@errmessage\errmessage
722     \let\errhelp@gobble
723     \def\errmessage##1{%
724       \xdef\@ExpectErrorMessage{##1}%
725     }%
726     \PackageError\TestPackageName{#1}\@ehc
727     \def\errhelp##1{%
728       \global\let\errhelp\saved@errhelp

```

```

729 }%
730 \global\let\@ResultErrorMessage\@empty
731 \def\errmessage##1{%
732   \xdef\@ResultErrorMessage{##1}%
733   \global\let\errmessage\saved@errmessage
734   % \Message{[ ##1}%
735   % \Message{] (ignored error)}%
736   % \Message{}%
737 }%
738 #2%
739 \endgroup
740 \Expect*{\@ResultErrorMessage}*{\@ExpectErrorMessage}%
741 }
742 \usepackage{scrfile}
743 \usepackage{hycolor}[2008/08/01]
744 \</test>

745 \<test1>
746 \begin{qstest}{NumNormalize}{num, normalize}
747   \def\test#1#2{%
748     \HyColor@NormalizeNum{#1}\cmd
749     \Expect*{\cmd}{#2}%
750   }%
751   \test{0}{0}%
752   \test{000}{0}%
753   \test{-1}{0}%
754   \test{ 0 }{0}%
755   \test{1.1}{1}%
756   \test{100}{1}%
757   \test{00100}{1}%
758   \test{99.99}{1}%
759   \test{0.0}{0}%
760   \test{00.00}{0}%
761   \test{0.}{0}%
762   \test{.0}{0}%
763   \test{0.1}{.1}%
764   \test{0.10}{.1}%
765   \test{0.1000}{.1}%
766   \test{.1000}{.1}%
767   \test{0.01}{.01}%
768   \test{0.01010}{.0101}%
769   \test{.0000000001}{.0000000001}%
770   \test{.9999999999}{.9999999999}%
771 \end{qstest}
772
773 \begin{qstest}{BookmarkColor without xcolor}{bookmark, noxcolor}
774   \def\test#1#2{%
775     \HyColor@BookmarkColor{#1}\cmd\TestPackageName\TestOptionName
776     \Expect*{\cmd}{#2}%
777   }%
778   \test{[rgb]{1,0,0}}{1 0 0}%
779   \test{[gray]{0.10}}{.1 .1 .1}%
780   \test{}{}%
781   \test{[rgb]{ 1 , 1 , 0 }}{1 1 0}%
782   \def\errortest[#1]#2{%
783     \ExpectError{%
784       Color model ‘#1’ is not supported\MessageBreak
785       without package ‘xcolor’ in\MessageBreak
786       ‘\TestOptionName=[#1]{#2}’% hash-ok
787     }{%
788       \test{[#1]{#2}}{}% hash-ok
789     }%
790   }%

```

```

791 \errortest[cmk]{1,0,0,0}%
792 \errortest[empty]{}%
793 \def\errortest#1{%
794   \ExpectError{%
795     This color specification is not supported\MessageBreak
796     without package 'xcolor' in\MessageBreak
797     '\TestOptionName=#1'%
798   }{%
799     \test{#1}{}%
800   }%
801 }%
802 \end{qstest}
803 </test1>

804 <*test1 | test2>
805 \begin{qstest}{X0134 without xcolor}{X0134, noxcolor}
806   \def\test#1#2{%
807     \HyColor@XZeroOneThreeFour{#1}\cmd\TestPackageName\TestOptionName
808     \Expect*{\cmd}{#2}%
809   }%
810   \test{[empty]{}{}%
811   \test{[rgb]{1,0,0}}{1 0 0}%
812   \test{[gray]{0.10}}{.1}%
813   \test{[cmk]{0,1,0,0}}{0 1 0 0}%
814   \test{[rgb]{ 1 , 1 , 0 }}{1 1 0}%
815   \def\errortest[#1]#2{%
816     \ExpectError{%
817       Color model '#1' is not supported\MessageBreak
818       without package 'xcolor' in\MessageBreak
819       'test-option=[#1]{#2}'% hash-ok
820     }{%
821       \HyColor@XZeroOneThreeFour{[#1]}{#2}\cmd
822       \TestPackageName\TestOptionName
823       \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
824     }%
825   }%
826   \errortest[Gray]{10}%
827   \errortest[cmk]{1,0,0}%
828   \def\errortest#1{%
829     \ExpectError{%
830       This color specification is not supported\MessageBreak
831       without package 'xcolor' in\MessageBreak
832       'test-option=#1'%
833     }{%
834       \HyColor@XZeroOneThreeFour{#1}\cmd\TestPackageName\TestOptionName
835       \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
836     }%
837   }%
838   \errortest{yellow}%
839 \end{qstest}
840

841 \begin{qstest}{HyperrefBorderColor without xcolor}%
842   {hyperref bordercolor, noxcolor}%
843   \def\test#1#2{%
844     \HyColor@HyperrefBorderColor{#1}\cmd\TestPackageName\TestOptionName
845     \Expect*{\cmd}{#2}%
846   }%
847   \test{[rgb]{1,0,0}}{1 0 0}%
848   \test{[gray]{0.10}}{.1 .1 .1}%
849   \test{[rgb]{ 1 , 1 , 0 }}{1 1 0}%
850   \def\errortest[#1]#2{%
851     \ExpectError{%
852       Color model '#1' is not supported\MessageBreak

```

```

853     without package 'xcolor' in\MessageBreak
854     'test-option=[#1]{#2}'% hash-ok
855 }{%
856   \HyColor@HyperrefBorderColor{[#1]{#2}}\cmd
857   \TestPackageName\TestOptionName
858   \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
859 }%
860 }%
861 \errortest[Gray]{10}%
862 \errortest[cmY]{1,0,0}%
863 \errortest[cmYk]{0,1,0,0}%
864 \def\errortest#1{%
865   \ExpectError{%
866     This color specification is not supported\MessageBreak
867     without package 'xcolor' in\MessageBreak
868     'test-option=#1'%
869   }{%
870     \HyColor@HyperrefBorderColor{#1}\cmd
871     \TestPackageName\TestOptionName
872     \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
873   }%
874 }%
875 \errortest{yellow}%
876 \end{qstest}
877 </test1 | test2>

878 <*test1 | test2>
879 \usepackage{xcolor}
880 \definecolor[named]{MyGreen}{rgb}{0,0.7,0}
881 \definecolor{mygreen}{named}{MyGreen}
882 </test1 | test2>

883 <*test1>
884 \begin{qstest}{BookmarkColor with xcolor}{bookmark, xcolor}
885   \def\test#1#2{%
886     \HyColor@BookmarkColor{#1}\cmd\PackageName\OptionName
887     \Expect*{\cmd}{#2}%
888   }%
889   \test{[rgb]{1,0,0}}{1 0 0}%
890   \test{[gray]{0.10}}{.1 .1 .1}%
891   \test{}{}%
892   \test{[rgb]{ 1 , 1 , 0 }}{1 1 0}%
893   \test{[cmYk]{1,0,0,0}}{0 1 1}%
894   \test{red}{1 0 0}%
895   \test{cyan}{0 1 1}%
896   \test{red!40!blue}{.4 0 .6}%
897   \test{[Gray]{10}}{.66667 .66667 .66667}%
898   \test{[RGB]{100,200,50}}{.39217 .78432 .19609}%
899   \test{[wave]{363}}{.00316 0 .00316}%
900   \test{[wave]814}{.00797 0 0}%
901   \test{[HSB]{100,200,50}}{.03473 .20833 .12152}%
902   \test{[HTML]{A800FF}}{.65881 0 1}%
903   \test{[cmY]{.3,.5,.2}}{.7 .5 .8}%
904   \test{[cmYk]{.3,.5,.2,.1}}{.6 .4 .7}%
905   \test{[hsb]{.3,.5,.2}}{.12 .2 .1}%
906   \test{[Hsb]{120,.5,.2}}{.1 .2 .1}%
907   \test{[tHsb]{120,.5,.2}}{.2 .2 .1}%
908   \test{[named]{MyGreen}}{0 .7 0}%
909   \test{mygreen}{0 .7 0}%
910 \end{qstest}
911
912 \begin{qstest}{HyperrefColor}{hyperref, color}
913   \def\test#1#2{%
914     \HyColor@HyperrefColor{#1}\cmd

```

```

915 \Expect*{\cmd}{#2}%
916 }%
917 \test{red}{red}%
918 \test{[rgb]{1,0,0}}{[rgb]{1,0,0}}%
919 \HyColor@HyperrefColor{\cmd
920 \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
921 \end{qstest}
922 </test1>

923 <*test1 | test2>
924 \begin{qstest}{X0134 with xcolor}{hyperref, X0134, xcolor}
925 \def\test#1#2{%
926 \HyColor@XZeroOneThreeFour{#1}\cmd\PackageName\OptionName
927 \Expect*{\cmd}{#2}%
928 }%
929 \test{[empty]{} }{}%
930 \test{[gray]{0.1}}{.1}%
931 \test{[rgb]{1,0.5,0.0}}{1 .5 0}%
932 \test{[cmyk]{0,1,0,0.5}}{0 1 0 .5}%
933 \test{[Gray]{10}}{.66667}%
934 \test{red}{1 0 0}%
935 \test{1 0 0}{1 0 0}%
936 \test{001.0 .23 0}{1 .23 0}%
937 \test{[named]{MyGreen}}{0 .7 0}%
938 \test{mygreen}{0 .7 0}%
939 \HyColor@XZeroOneThreeFour{\cmd\PackageName\OptionName
940 \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
941 \end{qstest}
942
943 \begin{qstest}{FieldColor}{hyperref, field, FieldColor}
944 \def\test#1#2{%
945 \HyColor@FieldColor{#1}\cmd\PackageName\OptionName
946 \Expect*{\cmd}{#2}%
947 }%
948 \test{}{}%
949 \test{[gray]{0.7}}{.7 g}%
950 \test{[rgb]{1,0,0}}{1 0 0 rg}%
951 \test{[cmyk]{0,1,0,0}}{0 1 0 0 k}%
952 \test{[cmy]{.5,.4,.3}}{.5 .6 .7 rg}%
953 \end{qstest}
954 </test1 | test2>

```

3.1 Test for package attachfile2

```

955 <*test3>
956 \def\atfi@SETRGBCOLORTest{set-rgb}
957 \def\atfi@SETGRAYCOLORTest{set-gray}
958 \def\atfi@SETCMYKCOLORTest{set-cmyk}
959 \def\Test#1#2#3#4#5{%
960 \begin{group}
961 \setbox0=\hbox{%
962 \begin{group}
963 \chardef\HyColor@PdfVersion=6 %
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}{#4}%
970 \end{group}
971 \begin{group}
972 \chardef\HyColor@PdfVersion=7 %
973 \HyColor@AttachfileColor{#1}\spec\inlinemacro\annot
974 \TestPackageName\TestOptionName

```

```

975         \edef\inline{\inlinemacro{test}}%
976         \expandafter\Expect\expandafter{\spec}{#2}%
977         \expandafter\Expect\expandafter{\inline}{#3}%
978         \expandafter\Expect\expandafter{\annot}{#5}%
979     \endgroup
980 }%
981 \Expect*{\the\wd0}{0.0pt}%
982 \endgroup
983 }
984 \newif\ifError
985 \def\TestError[#1]#2#3#4#5#6{%
986     \begingroup
987     \global\Errorfalse
988     \let\OrgPackageError\PackageError
989     \def\PackageError##1##2##3{%
990         \edef\TestTemp{##1}%
991         \ifx\TestTemp\TestPackageName
992             \Expect*{\ifError too many errors\else ok\fi}{ok}%
993             \Expect*{#6}*{##2}%
994             \global\Errortrue
995         \else
996             \OrgPackageError{##1}{##2}{##3}%
997         \fi
998     }%
999     \setbox0=\hbox{%
1000         \begingroup
1001         \chardef\HyColor@PdfVersion=#1 %
1002         \HyColor@AttachfileColor{#2}\spec\inlinemacro\annot
1003         \TestPackageName\TestOptionName
1004         \edef\inline{\inlinemacro{test}}%
1005         \expandafter\Expect\expandafter{\spec}{#3}%
1006         \expandafter\Expect\expandafter{\inline}{#4}%
1007         \expandafter\Expect\expandafter{\annot}{#5}%
1008         \endgroup
1009         \ifx\#6\%
1010             \else
1011             \Expect*{\ifError ok\else missing error\fi}{ok}%
1012         \fi
1013     }%
1014     \Expect*{\the\wd0}{0.0pt}%
1015 \endgroup
1016 }
1017 \def\NoEmptyModel{%
1018     Color model 'empty' is not permitted for option '\TestOptionName'%
1019 }
1020 \def\ModelNoXcolor#1#2{%
1021     Color model '#1' is not supported\MessageBreak
1022     without package 'xcolor' in\MessageBreak
1023     '\TestOptionName=[#1]{#2}'% hash-ok
1024 }
1025 \def\SpecNoXColor#1{%
1026     This color specification is not supported\MessageBreak
1027     without package 'xcolor' in\MessageBreak
1028     'test-option=#1'%
1029 }
1030 \begin{qstest}{\AttachfileColor}{\AttachfileColor}
1031     \Test{}{}{}{}{}%
1032     \Test{0.1 0.2 0.3}{[rgb]{.1,.2,.3}}{.1 .2 .3 set-rgb}%
1033         {/C[.1 .2 .3]}{/C[.1 .2 .3]}%
1034     \Test{[gray]{0.4}}{[gray]{0.4}}{.4 set-gray}%
1035         {/C[.4 .4 .4]}{/C[.4]}%
1036     \Test{[rgb]{0.3,.2,.1}}{[rgb]{0.3,.2,.1}}{.3 .2 .1 set-rgb}%

```



```

1096 \usepackage{scrfile}
1097 \usepackage{xcolor-patch}[2008/08/01]
1098 \usepackage{xcolor}
1099 </xcol2>
1100 <*xcol3>
1101 \usepackage{xcolor-patch}[2008/08/01]
1102 \usepackage{xcolor}
1103 \begin{document}
1104 </xcol3>
1105 \makeatletter
1106 \newcommand*{\ColModList}{%
1107   rgb,%
1108   cmy,%
1109   cmyk,%
1110   hsb,%
1111   Hsb,%
1112   tHsb,%
1113   gray,%
1114   RGB,%
1115   HTML,%
1116   HSB,%
1117   Gray,%
1118   % wave,
1119 }
1120 \newcommand*{\StartModel}{rgb}
1121 \newcommand*{\StartValues}{.1,.2,.3}
1122 \@for\x:=\ColModList\do{%
1123   \ifx\x\@empty
1124   \else
1125     \convertcolorspec\StartModel\StartValues\x\y
1126     \typeout{* [\StartModel]{\StartValues} ==> [\x]{\y}}%
1127     \@for\xx:=\ColModList\do{%
1128       \ifx\xx\@empty
1129       \else
1130         \convertcolorspec\x\y\xx\yy
1131         \typeout{* [\x]{\y} ==> [\xx]{\yy}}%
1132       \fi
1133     }%
1134   \fi
1135 }
1136 <xcol3>\end{document}
1137 <xcol1 | xcol2>\@@end
1138 </test - xcolor>

```

4 Installation

4.1 Download

Package. This package is available on CTAN¹:

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

[CTAN:macros/latex/contrib/oberdiek/hycolor.pdf](http://ctan.org/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](http://ctan.org/install/macros/latex/contrib/oberdiek.tds.zip)

TDS refers to the standard “A Directory Structure for T_EX Files” ([CTAN:tds/tds.pdf](http://ctan.org/tds/tds.pdf)). Directories with `texmf` in their name are usually organized this way.

¹ftp://ftp.ctan.org/tex-archive/

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. Package `attachfile2` comes with the Perl script `pdfatfi.pl` that should be installed in such a way that it can be called as `pdfatfi`. Example (linux):

```
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
```

4.3 Package installation

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

```
tex hycolor.dtx
```

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

<code>hycolor.sty</code>	\rightarrow <code>tex/latex/oberdiek/hycolor.sty</code>
<code>xcolor-patch.sty</code>	\rightarrow <code>tex/latex/oberdiek/xcolor-patch.sty</code>
<code>hycolor.pdf</code>	\rightarrow <code>doc/latex/oberdiek/hycolor.pdf</code>
<code>test/hycolor-test1.tex</code>	\rightarrow <code>doc/latex/oberdiek/test/hycolor-test1.tex</code>
<code>test/hycolor-test2.tex</code>	\rightarrow <code>doc/latex/oberdiek/test/hycolor-test2.tex</code>
<code>test/hycolor-test3.tex</code>	\rightarrow <code>doc/latex/oberdiek/test/hycolor-test3.tex</code>
<code>test/hycolor-test-xcol1.tex</code>	\rightarrow <code>doc/latex/oberdiek/test/hycolor-test-xcol1.tex</code>
<code>test/hycolor-test-xcol2.tex</code>	\rightarrow <code>doc/latex/oberdiek/test/hycolor-test-xcol2.tex</code>
<code>test/hycolor-test-xcol3.tex</code>	\rightarrow <code>doc/latex/oberdiek/test/hycolor-test-xcol3.tex</code>
<code>hycolor.dtx</code>	\rightarrow <code>source/latex/oberdiek/hycolor.dtx</code>

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

4.5 Some details for the interested

Attached source. The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is `pdftk`, e.g. unpack the file into the current directory:

```
pdftk hycolor.pdf unpack_files output .
```

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.

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
```

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.

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; numbers in roman refer to the code lines where the entry is used.

Symbols	
<code>\!</code>	<code>\@ResultErrorMessage</code> .. 730, 732, 740
<code>\"</code>	<code>\@ReturnAfterFi</code> 65, 70, <u>163</u>
<code>\+</code>	<code>\@car</code> 350
<code>\-</code>	<code>\@cclv</code> 630, 672
<code>\.</code>	<code>\@ehc</code> 136, 143, 299, 513, 524, 726
<code>\:</code>	<code>\@empty</code> .. 40, 50, 51, 111, 118, 125,
<code>\;</code>	185, 205, 213, 244, 285, 288,
<code>\></code>	290, 347, 406, 447, 449, 487,
<code>\@end</code>	498, 514, 516, 520, 730, 1123, 1128
<code>\@scl</code>	<code>\@expandtwoargs</code> 584, 598
630, 636, 641, 667, 672, 678, 683	<code>\@firstofone</code> 439, 537
<code>\@tmp</code>	<code>\@firstoftwo</code> 156, 169, 252, 307, 337, 377
558,	<code>\@for</code> 1122, 1127
563, 566, 583, 589, 592, 613,	<code>\@gobble</code> 436, 448, 515, 519, 722
614, 617, 618, 621, 622, 626,	<code>\@ifnextchar</code> 146, 192
627, 631, 632, 633, 637, 638,	<code>\@ifpackageloaded</code> 536, 539, 546
642, 646, 647, 654, 655, 659,	<code>\@one</code> 45
660, 663, 664, 668, 669, 673,	<code>\@nil</code> 41, 44, 49,
674, 675, 679, 680, 684, 688, 689	54, 61, 66, 78, 86, 101, 104, 123,
<code>\@ExpectErrorMessage</code>	151, 153, 158, 187, 194, 197,
724, 740	

218, 220, 235, 248, 259, 262, 272, 315, 320, 350, 389, 392, 410, 415, 433, 442, 455, 484, 495	\extractcolorspec . 121, 251, 413, 492
H	
\nnil 566, 578, 592, 606	\hbox 961, 999
\@onelevel@sanitize 32, 173, 175, 177, 179, 181	\Hy@pdfversion 429
\@secondoftwo 161, 167, 253, 309, 339, 380	\HyColor@@@UseColor 192, 197
\\ 46, 62, 1009	\HyColor@@@UseColor 192, 194
A	
\afterassignment 552	\HyColor@@IfRGB 315, 320
\AfterPackage 540	\HyColor@@UseColor 187, 191
\annot . . 964, 969, 973, 978, 1002, 1007	\HyColor@AttachfileColor 445, 964, 973, 1002
\AtBeginDocument 545	\HyColor@BookmarkColor . . 96, 775, 886
\atfi@SETCMYKCOLORTest 958	\HyColor@CheckDot 41, 44
\atfi@SETGRAYCOLORTest 957	\HyColor@CheckNum . . 324, 326, 328, 345
\atfi@SETRGBCOLORTest 956	\HyColor@DefSanitized . 6, 154, 155, 160
B	
\begin . 746, 773, 805, 841, 884, 912, 924, 943, 1030, 1056, 1084, 1103	\HyColor@DetectPdfVersion 424, 466, 478
C	
\catcode 7, 8, 9, 10, 11, 12, 13	\HyColor@dot 344, 355
\chardef 427, 429, 963, 972, 1001	\HyColor@ErrorModelNoXcolor 112, 131, 229, 239, 400, 488
\cmd 748, 749, 775, 776, 807, 808, 821, 823, 834, 835, 844, 845, 856, 858, 870, 872, 886, 887, 914, 915, 919, 920, 926, 927, 939, 940, 945, 946	\HyColor@ErrorSpecNoXcolor 126, 138, 277, 418, 499
\ColModList 1106, 1122, 1127	\HyColor@FieldBColor 283
\color 195, 198	\HyColor@FieldColor 284, 945
\convertcolorspec 99, 122, 223, 233, 264, 270, 387, 414, 482, 493, 1125, 1130	\HyColor@HyperrefBorderColor 384, 844, 856, 870
\csname 26, 166, 306, 426, 462, 474, 507	\HyColor@HyperrefColor . 200, 914, 919
D	
\definecolor 880, 881, 1052, 1053, 1054, 1055	\HyColor@IfModel 97, 145, 201, 211, 385, 480
\do 1122, 1127	\HyColor@IfRGB 247, 304, 409
\documentclass 704, 1090	\HyColor@IfXcolor . . 98, 120, 164, 222, 232, 250, 386, 412, 481, 491
E	
\end . . . 771, 802, 839, 876, 910, 921, 941, 953, 1050, 1081, 1085, 1136	\HyColor@MatchNum . . . 364, 368, 369
\endcsname 26, 166, 306, 426, 462, 474, 507	\HyColor@model . . 99, 103, 106, 133, 135, 154, 159, 202, 212, 214, 217, 219, 221, 223, 226, 233, 236, 252, 254, 257, 260, 263, 264, 268, 270, 273, 285, 292, 294, 296, 335, 370, 387, 391, 394, 458, 465, 477, 482, 510, 518
\errhelp 720, 722, 727, 728	\HyColor@model@cmymk 178, 179, 219, 260, 296, 477
\errmessage 721, 723, 731, 733	\HyColor@model@empty 172, 173, 212, 510
\Errorfalse 987	\HyColor@model@Gray 180, 181, 221, 263
\errortest 782, 791, 792, 793, 815, 826, 827, 828, 838, 850, 861, 862, 863, 864, 875	\HyColor@model@gray 106, 174, 175, 214, 224, 226, 254, 265, 268, 292, 394, 465
\Errortrue 994	\HyColor@model@rgb 100, 103, 122, 176, 177, 217, 234, 236, 257, 271, 273, 294, 335, 370, 388, 391, 414, 458, 483, 494
\Expect . . . 740, 749, 776, 808, 823, 835, 845, 858, 872, 887, 915, 920, 927, 940, 946, 967, 968, 969, 976, 977, 978, 981, 992, 993, 1005, 1006, 1007, 1011, 1014	\HyColor@NormalizeCommaCMYK 86, 220, 261
\ExpectError 718, 783, 794, 816, 829, 851, 865	\HyColor@NormalizeCommaRGB 78, 101, 104, 123, 218, 235, 248, 258, 272, 389, 392, 410, 415, 484, 495
	\HyColor@NormalizeNum 36, 79, 81, 83, 87, 89, 91, 93, 107, 215, 225, 255, 266, 395, 748
	\HyColor@one 343, 353
	\HyColor@PdfVersion 427, 429, 467, 479, 963, 972, 1001

\HyColor@resultfalse	332, 348, 357, 379, 451	\newcommand	712, 713, 715, 718, 1106, 1120, 1121
\HyColor@resulttrue	321, 376	\newif	304, 984
\HyColor@ReverseString	49, 54, 61, 66	\NoEmptyModel	1017, 1041, 1042, 1079, 1080
\HyColor@SpaceToComma	433, 455	\nofiles	1089
\HyColor@StripLeadingZeros	50, 71, 73		
\HyColor@temp	79, 80, 81, 82, 83, 84, 87, 88, 89, 90, 91, 92, 93, 94, 322, 323, 346, 347, 350, 351, 353, 355	O	
\HyColor@TwoSpaces	319, 323	\OptionName	886, 926, 939, 945
\HyColor@UseColor	182	\OrgPackageError	988, 996
\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	P	
\HyColor@WithModel	147, 153	\PackageError	132, 139, 299, 511, 522, 726, 988, 989
\HyColor@WithoutModel	149, 158	\PackageName	886, 926, 939, 945
\HyColor@XZeroOneThreeFour	210, 283, 286, 452, 807, 821, 834, 926, 939	\pdflastmatch	372, 373, 374
\HyColor@zero	342, 351	\pdfmatch	368, 700
		\ProvidesFile	696, 699, 702
		\ProvidesPackage	3, 535
I		R	
\ifdim	37	\rangeGray	641, 683
\ifError	984, 992, 1011	\rangeHSB	636, 678
\ifHyColor@result	304, 325, 327, 334, 453	\rangeRGB	625, 667
\ifin@	585, 599	\remove@to@nnil	561, 573, 587, 601
\ifnum	45, 368, 467, 479	\RequirePackage	5
\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, 559, 568, 571, 595, 650, 823, 835, 858, 872, 920, 940, 991, 1009, 1123, 1128	\reserved@a	542, 552, 557, 568, 582, 595, 610, 650
\immediate	716	S	
\in@	584, 598	\saved@errhelp	720, 728
\IncludeTests	707	\saved@errmessage	721, 733
\inline	966, 968, 975, 977, 1004, 1006	\setbox	961, 999
\inlinemacro	964, 966, 973, 975, 1002, 1004	\space	319, 368, 369
L		\spec	964, 967, 973, 976, 1002, 1005
\LogTests	708	\SpecNoXColor	1025, 1048, 1049
M		\StartModel	1120, 1125, 1126
\makeatletter	710, 1105	\StartValues	1121, 1125, 1126
\Message	715, 716, 734, 735, 736	\strip@prefix	372, 373, 374
\MessageBreak	133, 134, 140, 141, 784, 785, 795, 796, 817, 818, 830, 831, 852, 853, 866, 867, 1021, 1022, 1026, 1027	T	
\ModelNoXcolor	1020, 1045	\Test	959, 1031, 1032, 1034, 1036, 1038, 1040, 1057, 1058, 1060, 1062, 1064, 1066, 1067, 1068, 1069, 1070, 1072, 1073, 1075, 1076, 1077
N		\test	747, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 774, 778, 779, 780, 781, 788, 799, 806, 810, 811, 812, 813, 814, 843, 847, 848, 849, 885, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 913, 917, 918, 925, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 944, 948, 949, 950, 951, 952
\NeedsTeXFormat	2, 534, 1088	\TestError	985, 1041, 1042, 1043, 1046, 1048, 1049, 1079, 1080
		\TestOptionName	713, 775, 786, 797, 807, 822, 834, 844, 857, 871, 965, 974, 1003, 1018, 1023

