

The hycolor package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2007/04/11 v1.1

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	1
1.1	Summary	2
2	Implementation	2
2.1	Normalization	2
2.1.1	Sanitize value of color option	2
2.1.2	Normalize result	3
2.2	Main algorithm for color options	5
2.3	Package bookmark	5
2.4	Utils	7
2.5	Package hyperref	7
2.5.1	Options Hyp.*color	7
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
3	Test	14
4	Installation	18
4.1	Download	18
4.2	Bundle installation	19
4.3	Package installation	19
4.4	Refresh file name databases	19
4.5	Some details for the interested	19
5	History	20
	[2007/04/09 v1.0]	20
	[2007/04/11 v1.1]	20
6	Index	20

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

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

2 Implementation

```

1 (*package)
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{hycolor}%
4 [2007/04/11 v1.1 Code for color options of hyperref/bookmark (HO)]%

```

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
5 \begingroup
6 \catcode'\!=13 %
7 \catcode'\:=13 %
8 \catcode'\-=13 %
9 \catcode'\+=13 %
10 \catcode'\;=13 %
11 \catcode'\ "=13 %
12 \catcode'\>=13 %
13 \edef\x{%
14 \def\noexpand!\string!}%
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 }%
22 \def\y#1{\endgroup
23 \def\HyColor@DefSanitized##1##2{%
24 \begingroup
25 \csname @safe@activetrue\endcsname
26 #1%
27 \edef\x{\endgroup
28 \def\noexpand##1{##2}%
29 }%
30 \x
31 \@onelevel@sanitize##1%
32 }%
33 }%
34 \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
35 \def\HyColor@NormalizeNum#1#2{%
36 \ifdim#1pt<\z@
```

```

37   \def#2{0}%
38   \else
39   \edef#2{\zap@space#1 \@empty}%
40   \expandafter\HyColor@CheckDot#2..\@nil#2%
41   \fi
42 }
43 \def\HyColor@CheckDot#1.#2.#3\@nil#4{%
44   \ifnum0#1<\@ne
45   \ifx\#2\%
46   \def#4{0}%
47   \else
48   \edef#4{\HyColor@ReverseString#2\@nil{}}%
49   \edef#4{\expandafter\HyColor@StripLeadingZeros#4\@empty}%
50   \ifx#4\@empty
51   \def#4{0}%
52   \else
53   \edef#4{.\expandafter\HyColor@ReverseString#4\@nil{}}%
54   \fi
55   \fi
56   \else
57   \def#4{1}%
58   \fi
59 }
60 \def\HyColor@ReverseString#1#2\@nil#3{%
61   \ifx\#2\%
62   #1#3%
63   \else
64   \@ReturnAfterFi{%
65     \HyColor@ReverseString#2\@nil{#1#3}%
66   }%
67   \fi
68 }
69 \long\def\@ReturnAfterFi#1\fi{\fi#1}
70 \def\HyColor@StripLeadingZeros#1{%
71   \ifx#10%
72   \expandafter\HyColor@StripLeadingZeros
73   \else
74   #1%
75   \fi
76 }

```

\HyColor@NormalizeCommaRGB

```

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

```

\HyColor@NormalizeCommaCMYK

```

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

```

2.2 Main algorithm for color options

Procedure Main color option algorithm

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:

`rgbcolor=r g b` \equiv `color=[rgb]{r,g,b}`

Package bookmark stores the result in macro `\BKM@color`. The empty string is interpreted as *no color*.

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 \leftarrow no color;

end

case *with model*

if *with xcolor* **then**

cmd \leftarrow ConvertToRGB(*model*, *values*);

else

if *model* = *rgb* **then**

cmd \leftarrow *values* as normalized values;

else if *model* = *gray* **then**

cmd \leftarrow *values* as normalized tripled values;

else

 error;

end

end

end

otherwise

if *with xcolor* **then**

 (*model*, *values* \leftarrow get model and values;

cmd \leftarrow ConvertToRGB(*model*, *values*);

else

 error;

end

end

end

```

95 \def\HyColor@BookmarkColor#1#2#3#4{%
96   \HyColor@IfModel{#1}{%
97     \HyColor@IfXcolor{%
98       \convertcolorspec\HyColor@model\HyColor@values
99       \HyColor@model@rgb#2%
100     \expandafter\HyColor@NormalizeCommaRGB#2\@nil#2%
```

```

101 }{%
102   \ifx\HyColor@model\HyColor@model@rgb
103     \expandafter\HyColor@NormalizeCommaRGB\HyColor@values\@nil#2%
104   \else
105     \ifx\HyColor@model\HyColor@model@gray
106       \expandafter\HyColor@NormalizeNum
107       \expandafter{\HyColor@values}#2%
108     \edef#2{#2 #2 #2}%
109   \else
110     \let#2\@empty
111     \HyColor@ErrorModelNoXcolor{#3}{#4}%
112   \fi
113 \fi
114 }%
115 }{%
116   \let#2\HyColor@values
117   \ifx#2\@empty
118   \else
119     \HyColor@IfXcolor{%
120       \extractcolorspec{#1}#2%
121       \expandafter\convertcolorspec#2\HyColor@model@rgb#2%
122       \expandafter\HyColor@NormalizeCommaRGB#2\@nil#2%
123     }{%
124       \let#2\@empty
125       \HyColor@ErrorSpecNoXcolor{#3}{#4}%
126     }%
127   \fi
128 }%
129 }

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

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

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

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

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

```

2.4 Utils

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

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

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

2.5 Package hyperref

2.5.1 Options Hyp.*color

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

Procedure HyperrefColor(*value*, *cmd*)

Param: *value* (value of the option)

Param: *cmd* (macro for result)

```
switch value do
  case empty
    cmd  $\leftarrow$  no color;
  end
  case with model
    Call \color with model;
  end
  case without model
    Call \color without model;
  end
end
end
```

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


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
  else
    error(package, option, "Missing xcolor"), cmd ← no color;
  end
end
end
```

\HyColor@XZeroOneThreeFour

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

```

270         \HyColor@model@rgb#2%
271         \expandafter\HyColor@NormalizeCommaRGB#2\@nil#2%
272         \let\HyColor@model\HyColor@model@rgb
273         \fi\fi\fi\fi
274     }{%
275         \let#2\relax
276         \HyColor@ErrorSpecNoXcolor{#3}{#4}%
277     }%
278 }%
279 \fi
280 }%
281 }

```

2.5.3 Field options

\HyColor@FieldBColor

```

282 \let\HyColor@FieldBColor\HyColor@XZeroOneThreeFour

```

\HyColor@FieldColor

```

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

```

2.5.4 Detection for naked RGB values

\HyColor@IfRGB

```

303 \begingroup\expandafter\expandafter\expandafter\endgroup
304 \expandafter\ifx\csname pdfmatch\endcsname\relax
305     \expandafter\@firstoftwo
306 \else
307     \expandafter\@secondoftwo
308 \fi
309 {%
310     \newif\ifHyColor@result
311     \begingroup
312         \def\x#1{\endgroup
313             \def\HyColor@IfRGB##1{%
314                 \HyColor@@IfRGB##1#1#1\@nil
315             }%
316         }%
317     \x{ }%
318     \edef\HyColor@TwoSpaces{\space\space}%
319     \def\HyColor@@IfRGB#1 #2 #3 #4\@nil{%
320         \HyColor@resulttrue
321         \def\HyColor@temp{#4}%

```

```

322 \ifx\HyColor@temp\HyColor@TwoSpaces
323 \HyColor@CheckNum{#1}%
324 \ifHyColor@result
325 \HyColor@CheckNum{#2}%
326 \ifHyColor@result
327 \HyColor@CheckNum{#3}%
328 \fi
329 \fi
330 \else
331 \HyColor@resultfalse
332 \fi
333 \ifHyColor@result
334 \let\HyColor@model\HyColor@model@rgb
335 \edef\HyColor@values{#1,#2,#3}%
336 \expandafter\@firstoftwo
337 \else
338 \expandafter\@secondoftwo
339 \fi
340 }%
341 \def\HyColor@zero{0}%
342 \def\HyColor@one{1}%
343 \def\HyColor@dot{.}%
344 \def\HyColor@CheckNum#1{%
345 \def\HyColor@temp{#1}%
346 \ifx\HyColor@temp\@empty
347 \HyColor@resultfalse
348 \else
349 \edef\HyColor@temp{\@car#1\@nil}%
350 \ifx\HyColor@temp\HyColor@zero
351 \else
352 \ifx\HyColor@temp\HyColor@one
353 \else
354 \ifx\HyColor@temp\HyColor@dot
355 \else
356 \HyColor@resultfalse
357 \fi
358 \fi
359 \fi
360 \fi
361 }%
362 }{%
363 \def\HyColor@MatchNum{%
364 (0*1|string\.0*|0*1|0+\string\.[0-9]*|\string\.[0-9]+)%
365 }%
366 \def\HyColor@IfRGB#1{%
367 \ifnum\pdfmatch{^{\HyColor@MatchNum\space\HyColor@MatchNum
368 \space\HyColor@MatchNum$}{#1}>\z@
369 \let\HyColor@model\HyColor@model@rgb
370 \edef\HyColor@values{%
371 \expandafter\strip@prefix\pdfmatch1,%
372 \expandafter\strip@prefix\pdfmatch2,%
373 \expandafter\strip@prefix\pdfmatch3%
374 }%
375 \expandafter\@firstoftwo
376 \else
377 \expandafter\@secondoftwo
378 \fi
379 }%
380 }

```

2.5.5 Options ***bordercolor**

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

```

`\HyColor@HyperrefBorderColor`

```

381 \def\HyColor@HyperrefBorderColor#1#2#3#4{%
382   \HyColor@IfModel{#1}{%
383     \HyColor@IfXcolor{%
384       \convertcolourspec\HyColor@model\HyColor@values
385         \HyColor@model@rgb#2%
386       \expandafter\HyColor@NormalizeCommaRGB#2\@nil#2%
387     }{%
388       \ifx\HyColor@model\HyColor@model@rgb
389         \expandafter\HyColor@NormalizeCommaRGB\HyColor@values\@nil#2%
390       \else
391         \ifx\HyColor@model\HyColor@model@gray
392           \expandafter\HyColor@NormalizeNum
393             \expandafter{\HyColor@values}#2%
394         \edef#2{#2 #2 #2}%
395       \else
396         \let#2\relax
397         \HyColor@ErrorModelNoXcolor{#3}{#4}%
398       \fi

```

```

399     \fi
400   }%
401 }{%
402   \let#2\HyColor@values
403   \ifx#2\@empty
404     \let#2\relax
405   \else
406     \expandafter\HyColor@IfRGB\expandafter{\HyColor@values}{%
407       \expandafter\HyColor@NormalizeCommaRGB\HyColor@values\@nil#2%
408     }{%
409       \HyColor@IfXcolor{%
410         \extractcolorspec{#1}#2%
411         \expandafter\convertcolorspec#2\HyColor@model@rgb#2%
412         \expandafter\HyColor@NormalizeCommaRGB#2\@nil#2%
413       }{%
414         \let#2\relax
415         \HyColor@ErrorSpecNoXcolor{#3}{#4}%
416       }%
417     }%
418   \fi
419 }%
420 }

421 </package>

```

3 Test

```

]

422 <*test1>
423 \ProvidesFile{hycolor-test1.tex}[2007/04/11 test file 1]
424 </test1>

425 <*test2>
426 \ProvidesFile{hycolor-test2.tex}[2007/04/11 test file 2]
427 \let\pdfmatch\relax
428 </test2>

429 <*test>

430 \documentclass{article}
431
432 \usepackage{qstest}
433 \IncludeTests{*}
434 \LogTests{log}{*}{*}
435
436 \makeatletter
437
438 \newcommand*\TestPackageName{test-package}
439 \newcommand*\TestOptionName{test-option}
440
441 \newcommand\Message{}
442 \def\Message#1#{\immediate\write16}
443
444 \newcommand*\ExpectError}[2]{%
445   \begingroup
446     \global\let\saved@errhelp\errhelp
447     \global\let\saved@errmessage\errmessage
448     \let\errhelp\@gobble
449     \def\errmessage##1{%
450       \xdef\ExpectErrorMessage{##1}%
451     }%
452     \PackageError\TestPackageName{#1}\@ehc
453     \def\errhelp##1{%

```

```

454 \global\let\errhelp\saved@errhelp
455 }%
456 \global\let\@ResultErrorMessage\@empty
457 \def\errmessage##1{%
458 \xdef\@ResultErrorMessage{##1}%
459 \global\let\errmessage\saved@errmessage
460 % \Message{[ ##1}%
461 % \Message{] (ignored error)}%
462 % \Message{}%
463 }%
464 #2%
465 \endgroup
466 \Expect*{\@ResultErrorMessage}*{\@ExpectErrorMessage}%
467 }
468
469 \usepackage{hycolor}
470 /test)

471 (*test1)
472 \begin{qstest}{NumNormalize}{num, normalize}
473 \def\test#1#2{%
474 \HyColor@NormalizeNum{#1}\cmd
475 \Expect*{\cmd}{#2}%
476 }%
477 \test{0}{0}%
478 \test{000}{0}%
479 \test{-1}{0}%
480 \test{ 0 }{0}%
481 \test{1.1}{1}%
482 \test{100}{1}%
483 \test{00100}{1}%
484 \test{99.99}{1}%
485 \test{0.0}{0}%
486 \test{00.00}{0}%
487 \test{0.}{0}%
488 \test{.0}{0}%
489 \test{0.1}{.1}%
490 \test{0.10}{.1}%
491 \test{0.1000}{.1}%
492 \test{.1000}{.1}%
493 \test{0.01}{.01}%
494 \test{0.01010}{.0101}%
495 \test{.0000000001}{.0000000001}%
496 \test{.9999999999}{.9999999999}%
497 \end{qstest}
498
499 \begin{qstest}{BookmarkColor without xcolor}{bookmark, noxcolor}
500 \def\test#1#2{%
501 \HyColor@BookmarkColor{#1}\cmd\TestPackageName\TestOptionName
502 \Expect*{\cmd}{#2}%
503 }%
504 \test{[rgb]{1,0,0}}{1 0 0}%
505 \test{[gray]{0.10}}{.1 .1 .1}%
506 \test{}{}%
507 \test{[rgb]{ 1 , 1 , 0 }}{1 1 0}%
508 \def\errortest[#1]#2{%
509 \ExpectError{
510 Color model '#1' is not supported\MessageBreak
511 without package 'xcolor' in\MessageBreak
512 '\TestOptionName=[#1]{#2}'% hash-ok
513 }%
514 \test{[#1]{#2}}{}% hash-ok
515 }%

```

```

516 }%
517 \errortest[cmk]{1,0,0,0}%
518 \errortest[empty]{}%
519 \def\errortest#1{%
520   \ExpectError{%
521     This color specification is not supported\MessageBreak
522     without package 'xcolor' in\MessageBreak
523     '\TestOptionName=#1'%
524   }{%
525     \test{#1}{}%
526   }%
527 }%
528 \end{qstest}
529 \test1)

530 (*test1 | test2)
531 \begin{qstest}{X0134 without xcolor}{X0134, noxcolor}
532   \def\test#1#2{%
533     \HyColor@XZeroOneThreeFour{#1}\cmd\TestPackageName\TestOptionName
534     \Expect*{\cmd}{#2}%
535   }%
536   \test{[empty]{}{}}{%
537     \test{[rgb]{1,0,0}}{1 0 0}%
538     \test{[gray]{0.10}}{.1}%
539     \test{[cmk]{0,1,0,0}}{0 1 0 0}%
540     \test{[rgb]{ 1 , 1 , 0 }}{1 1 0}%
541     \def\errortest[#1]#2{%
542       \ExpectError{%
543         Color model '#1' is not supported\MessageBreak
544         without package 'xcolor' in\MessageBreak
545         'test-option=[#1]{#2}'% hash-ok
546       }{%
547         \HyColor@XZeroOneThreeFour{[#1]{#2}}\cmd
548         \TestPackageName\TestOptionName
549         \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
550       }%
551     }%
552     \errortest[Gray]{10}%
553     \errortest[cmk]{1,0,0}%
554     \def\errortest#1{%
555       \ExpectError{%
556         This color specification is not supported\MessageBreak
557         without package 'xcolor' in\MessageBreak
558         'test-option=#1'%
559       }{%
560         \HyColor@XZeroOneThreeFour{#1}\cmd\TestPackageName\TestOptionName
561         \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
562       }%
563     }%
564     \errortest{yellow}%
565   \end{qstest}
566
567 \begin{qstest}{HyperrefBorderColor without xcolor}%
568   {hyperref bordercolor, noxcolor}%
569   \def\test#1#2{%
570     \HyColor@HyperrefBorderColor{#1}\cmd\TestPackageName\TestOptionName
571     \Expect*{\cmd}{#2}%
572   }%
573   \test{[rgb]{1,0,0}}{1 0 0}%
574   \test{[gray]{0.10}}{.1 .1 .1}%
575   \test{[rgb]{ 1 , 1 , 0 }}{1 1 0}%
576   \def\errortest[#1]#2{%
577     \ExpectError{%

```



```

578     Color model '#1' is not supported\MessageBreak
579     without package 'xcolor' in\MessageBreak
580     'test-option=[#1]{#2}'% hash-ok
581 }{%
582   \HyColor@HyperrefBorderColor{[#1]{#2}}\cmd
583   \TestPackageName\TestOptionName
584   \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
585 }%
586 }%
587 \errortest[Gray]{10}%
588 \errortest[cmY]{1,0,0}%
589 \errortest[cmYk]{0,1,0,0}%
590 \def\errortest#1{%
591   \ExpectError{%
592     This color specification is not supported\MessageBreak
593     without package 'xcolor' in\MessageBreak
594     'test-option=#1'%
595   }{%
596     \HyColor@HyperrefBorderColor{#1}\cmd
597     \TestPackageName\TestOptionName
598     \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
599   }%
600 }%
601 \errortest{yellow}%
602 \end{qstest}
603 </test1 | test2>

604 <*test1 | test2>
605 \usepackage{xcolor}
606 \definecolor[named]{MyGreen}{rgb}{0,0.7,0}
607 \definecolor{mygreen}{named}{MyGreen}
608 </test1 | test2>

609 <*test1>
610 \begin{qstest}{BookmarkColor with xcolor}{bookmark, xcolor}
611   \def\test#1#2{%
612     \HyColor@BookmarkColor{#1}\cmd\PackageName\OptionName
613     \Expect*{\cmd}{#2}%
614   }%
615   \test{[rgb]{1,0,0}}{1 0 0}%
616   \test{[gray]{0.10}}{.1 .1 .1}%
617   \test{}{}%
618   \test{[rgb]{ 1 , 1 , 0 }}{1 1 0}%
619   \test{[cmYk]{1,0,0,0}}{0 1 1}%
620   \test{red}{1 0 0}%
621   \test{cyan}{0 1 1}%
622   \test{red!40!blue}{.4 0 .6}%
623   \test{[Gray]{10}}{.66667 .66667 .66667}%
624   \test{[RGB]{100,200,50}}{.39217 .78432 .19609}%
625   \test{[wave]{363}}{.00316 0 .00316}%
626   \test{[wave]814}{.00797 0 0}%
627   \test{[HSB]{100,200,50}}{.03473 .20833 .12152}%
628   \test{[HTML]{A800FF}}{.65881 0 1}%
629   \test{[cmY]{.3,.5,.2}}{.7 .5 .8}%
630   \test{[cmYk]{.3,.5,.2,.1}}{.6 .4 .7}%
631   \test{[hsb]{.3,.5,.2}}{.12 .2 .1}%
632   \test{[Hsb]{120,.5,.2}}{.1 .2 .1}%
633   \test{[tHsb]{120,.5,.2}}{.2 .2 .1}%
634   \test{[named]{MyGreen}}{0 .7 0}%
635   \test{mygreen}{0 .7 0}%
636 \end{qstest}
637
638 \begin{qstest}{HyperrefColor}{hyperref, color}
639   \def\test#1#2{%

```

```

640 \HyColor@HyperrefColor{#1}\cmd
641 \Expect*{\cmd}{#2}%
642 }%
643 \test{red}{red}%
644 \test{[rgb]{1,0,0}}{[rgb]{1,0,0}}%
645 \HyColor@HyperrefColor{\cmd
646 \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
647 \end{qstest}
648 \test1)

649 (*test1 | test2)
650 \begin{qstest}{X0134 with xcolor}{hyperref, X0134, xcolor}
651 \def\test#1#2{%
652 \HyColor@XZeroOneThreeFour{#1}\cmd\PackageName\OptionName
653 \Expect*{\cmd}{#2}%
654 }%
655 \test{[empty]}{}{}%
656 \test{[gray]{0.1}}{.1}%
657 \test{[rgb]{1,0.5,0.0}}{1 .5 0}%
658 \test{[cmyk]{0,1,0,0.5}}{0 1 0 .5}%
659 \test{[Gray]{10}}{.66667}%
660 \test{red}{1 0 0}%
661 \test{1 0 0}{1 0 0}%
662 \test{001.0 .23 0}{1 .23 0}%
663 \test{[named]{MyGreen}}{0 .7 0}%
664 \test{mygreen}{0 .7 0}%
665 \HyColor@XZeroOneThreeFour{\cmd\PackageName\OptionName
666 \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
667 \end{qstest}
668
669 \begin{qstest}{FieldColor}{hyperref, field, FieldColor}
670 \def\test#1#2{%
671 \HyColor@FieldColor{#1}\cmd\PackageName\OptionName
672 \Expect*{\cmd}{#2}%
673 }%
674 \test{}{}%
675 \test{[gray]{0.7}}{.7 g}%
676 \test{[rgb]{1,0,0}}{1 0 0 rg}%
677 \test{[cmyk]{0,1,0,0}}{0 1 0 0 k}%
678 \test{[cmy]{.5,.4,.3}}{.5 .6 .7 rg}%
679 \end{qstest}
680 \test1 | test2)

681 (*test)
682 \begin{document}
683 \end{document}
684 \test)

```

4 Installation

4.1 Download

Package. This package is available on CTAN¹:

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

[CTAN:macros/latex/contrib/oberdiek/hycolor.pdf](http://ftp.ctan.org/tex-archive/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.

¹<http://ftp.ctan.org/tex-archive/>

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

TDS refers to the standard “A Directory Structure for \TeX 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. 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>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>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.

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	
\!	6
\"	11
\+	9
\-	8
\.	364
\:	7
\;	10
\>	12
\@ExpectErrorMessage	450, 466
\@ResultErrorMessage	456, 458, 466
\@ReturnAfterFi	64, 69, <u>162</u>
\@car	349
\@ehc	135, 142, 298, 452
\@empty	39, 49, 50, 110, 117, 124, 184, 204, 212, 243, 284, 287, 289, 346, 403, 456
\@firstoftwo	155, 168, 251, 305, 336, 375
\@gobble	448
\@ifnextchar	145, 191
\@ne	44
\@nil	40, 43, 48, 53, 60, 65, 77, 85, 100, 103, 122, 150, 152, 157, 186, 193, 196, 217, 219, 234, 247, 258, 261, 271, 314, 319, 349, 386, 389, 407, 412
\@onelevel@sanitize	31, 172, 174, 176, 178, 180
\@secondoftwo	160, 166, 252, 307, 338, 377
\%	45, 61
B	
\begin	472, 499, 531, 567, 610, 638, 650, 669, 682
C	
\catcode	6, 7, 8, 9, 10, 11, 12
\cmd	474, 475, 501, 502, 533, 534, 547, 549, 560, 561, 570, 571, 582, 584, 596, 598, 612, 613, 640, 641, 645, 646, 652, 653, 665, 666, 671, 672
\color	194, 197
\convertcolorspec	98, 121, 222, 232, 263, 269, 384, 411
\csname	25, 165, 304
D	
\definecolor	606, 607
\documentclass	430
E	
\end	497, 528, 565, 602, 636, 647, 667, 679, 683
\endcsname	25, 165, 304
\errhelp	446, 448, 453, 454
\errmessage	447, 449, 457, 459

\errortest	508, 517, 518, 519, 541, 552, 553, 554, 564, 576, 587, 588, 589, 590, 601
\Expect	466, 475, 502, 534, 549, 561, 571, 584, 598, 613, 641, 646, 653, 666, 672
\ExpectError	444, 509, 520, 542, 555, 577, 591
\extractcolorspec	120, 250, 410
H	
\HyColor@@@UseColor	191, 196
\HyColor@@@UseColor	191, 193
\HyColor@@IfRGB	314, 319
\HyColor@@UseColor	186, 190
\HyColor@BookmarkColor	95, 501, 612
\HyColor@CheckDot	40, 43
\HyColor@CheckNum	323, 325, 327, 344
\HyColor@DefSanitized	5, 153, 154, 159
\HyColor@dot	343, 354
\HyColor@ErrorModelNoXcolor	111, 130, 228, 238, 397
\HyColor@ErrorSpecNoXcolor	125, 137, 276, 415
\HyColor@FieldBColor	282
\HyColor@FieldColor	283, 671
\HyColor@HyperrefBorderColor	381, 570, 582, 596
\HyColor@HyperrefColor	199, 640, 645
\HyColor@IfModel	96, 144, 200, 210, 382
\HyColor@IfRGB	246, 303, 406
\HyColor@IfXcolor	97, 119, 163, 221, 231, 249, 383, 409
\HyColor@MatchNum	363, 367, 368
\HyColor@model	98, 102, 105, 132, 134, 153, 158, 201, 211, 213, 216, 218, 220, 222, 225, 232, 235, 251, 253, 256, 259, 262, 263, 267, 269, 272, 284, 291, 293, 295, 334, 369, 384, 388, 391
\HyColor@model@cmyk	177, 178, 218, 259, 295
\HyColor@model@empty	171, 172, 211
\HyColor@model@Gray	179, 180, 220, 262
\HyColor@model@gray	105, 173, 174, 213, 223, 225, 253, 264, 267, 291, 391
\HyColor@model@rgb	99, 102, 121, 175, 176, 216, 233, 235, 256, 270, 272, 293, 334, 369, 385, 388, 411
\HyColor@NormalizeCommaCMYK	85, 219, 260
\HyColor@NormalizeCommaRGB	77, 100, 103, 122, 217, 234, 247, 257, 271, 386, 389, 407, 412
\HyColor@NormalizeNum	35, 78, 80, 82, 86, 88, 90, 92, 106, 214, 224, 254, 265, 392, 474
\HyColor@one	342, 352
\HyColor@resultfalse	331, 347, 356
\HyColor@resulttrue	320
\HyColor@ReverseString	48, 53, 60, 65
\HyColor@StripLeadingZeros	49, 70, 72
\HyColor@temp	78, 79, 80, 81, 82, 83, 86, 87, 88, 89, 90, 91, 92, 93, 321, 322, 345, 346, 349, 350, 352, 354
\HyColor@TwoSpaces	318, 322
\HyColor@UseColor	181
\HyColor@values	98, 103, 107, 116, 134, 141, 154, 159, 201, 203, 215, 217, 219, 222, 232, 242, 246, 247, 250, 252, 255, 258, 261, 263, 266, 269, 335, 370, 384, 389, 393, 402, 406, 407
\HyColor@WithModel	146, 152
\HyColor@WithoutModel	148, 157
\HyColor@XZeroOneThreeFour	209, 282, 285, 533, 547, 560, 652, 665
\HyColor@zero	341, 350
I	
\ifdim	36
\ifHyColor@result	310, 324, 326, 333
\ifnum	44, 367
\ifx	45, 50, 61, 71, 102, 105, 117, 165, 182, 184, 204, 211, 213, 216, 218, 220, 243, 253, 256, 259, 262, 286, 289, 291, 293, 295, 304, 322, 346, 350, 352, 354, 388, 391, 403, 549, 561, 584, 598, 646, 666
\immediate	442
\IncludeTests	433
L	
\LogTests	434
M	
\makeatletter	436
\Message	441, 442, 460, 461, 462
\MessageBreak	132, 133, 139, 140, 510, 511, 521, 522, 543, 544, 556, 557, 578, 579, 592, 593
N	
\NeedsTeXFormat	2
\newcommand	438, 439, 441, 444
\newif	310
O	
\OptionName	612, 652, 665, 671
P	
\PackageError	131, 138, 298, 452
\PackageName	612, 652, 665, 671
\pdflastmatch	371, 372, 373
\pdfmatch	367, 427
\ProvidesFile	423, 426
\ProvidesPackage	3
S	
\saved@errhelp	446, 454
\saved@errmessage	447, 459
\space	318, 367, 368
\strip@prefix	371, 372, 373

T		501, 533, 548, 560, 570, 583, 597
<code>\test</code>	473, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 500, 504, 505, 506, 507, 514, 525, 532, 536, 537, 538, 539, 540, 569, 573, 574, 575, 611, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 639, 643, 644, 651, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 670, 674, 675, 676, 677, 678	
<code>\TestOptionName</code>	... 439, 501, 512, 523, 533, 548, 560, 570, 583, 597	
<code>\TestPackageName</code> 438, 452,	
U		
<code>\usepackage</code>	432, 469, 605
W		
<code>\write</code>	442
X		
<code>\x</code>	13, 27, 30, 34, 312, 317
Y		
<code>\y</code>	22, 34
Z		
<code>\z@</code>	36, 368
<code>\zap@space</code>	39