

# The hycolor package

Heiko Oberdiek  
<oberdiek@uni-freiburg.de>

2008/07/29 v1.2

## 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 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
<b>3</b>	<b>Test</b>	<b>16</b>
3.1	Test for package attachfile2	20
<b>4</b>	<b>Installation</b>	<b>23</b>
4.1	Download	23
4.2	Bundle installation	23
4.3	Package installation	23
4.4	Refresh file name databases	23
4.5	Some details for the interested	24
<b>5</b>	<b>History</b>	<b>24</b>
	[2007/04/09 v1.0]	24
	[2007/04/11 v1.1]	24
	[2008/07/29 v1.2]	24
<b>6</b>	<b>Index</b>	<b>24</b>

# 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 [2008/07/29 v1.2 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** 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

```

---

```

95 \def\HyColor@BookmarkColor#1#2#3#4{%
96   \HyColor@IfModel{#1}{%
97     \HyColor@IfXcolor{%
98       \convertcolourspec\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 ← no color;
  end
  case with model
    Call \color with model;
  end
  case without model
    Call \color without model;
  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@cmyk
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 \newif\ifHyColor@result
304 \begingroup\expandafter\expandafter\expandafter\endgroup
305 \expandafter\ifx\csname pdfmatch\endcsname\relax
306   \expandafter\@firstoftwo
307 \else
308   \expandafter\@secondoftwo
309 \fi
310 {%
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 \HyColor@resulttrue
376 \expandafter\@firstoftwo
377 \else
378 \HyColor@resultfalse
379 \expandafter\@secondoftwo
380 \fi
381 }%
382 }

```

### 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 ← 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`

```

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

```

```

401     \fi
402   }%
403 }{%
404   \let#2\HyColor@values
405   \ifx#2\@empty
406     \let#2\relax
407   \else
408     \expandafter\HyColor@IfRGB\expandafter{\HyColor@values}{%
409       \expandafter\HyColor@NormalizeCommaRGB\HyColor@values\@nil#2%
410     }{%
411       \HyColor@IfXcolor{%
412         \extractcolorspec{#1}#2%
413         \expandafter\convertcolorspec#2\HyColor@model@rgb#2%
414         \expandafter\HyColor@NormalizeCommaRGB#2\@nil#2%
415       }{%
416         \let#2\relax
417         \HyColor@ErrorSpecNoXcolor{#3}{#4}%
418       }%
419     }%
420   \fi
421 }%
422 }

```

## 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

```

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

```

\HyColor@SpaceToComma

```

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

```

\HyColor@AttachfileColor

```
444 \def\HyColor@AttachfileColor#1#2#3#4#5#6{%
445   \def#2{#1}%
446   \ifx#2\@empty
447     \let#3\@gobble
448     \let#4\@empty
449   \else
450     \HyColor@resultfalse
451     \HyColor@XZeroOneThreeFour{#1}#3{#5}{#6}%
452     \ifHyColor@result
453       \edef#2{%
454         [rgb]{\expandafter\HyColor@SpaceToComma#3 \@nil}%
455       }%
456     \fi
457     \ifx\HyColor@model\HyColor@model@rgb
458       \edef#4{/C[#3]}% hash-ok
459       \edef#3##1{%
460         #3 %
461         \noexpand\csname atfi@SETRGBCOLOR##1\noexpand\endcsname
462       }%
463     \else
464       \ifx\HyColor@model\HyColor@model@gray
465         \HyColor@DetectPdfVersion
466         \ifnum\HyColor@PdfVersion<7 %
467           \edef#4{/C[#3 #3 #3]}% hash-ok
468         \else
469           \edef#4{/C[#3]}% hash-ok
470         \fi
471         \edef#3##1{%
472           #3 %
473           \noexpand\csname atfi@SETGRAYCOLOR##1\noexpand\endcsname
474         }%
475       \else
476         \ifx\HyColor@model\HyColor@model@cmyk
477           \HyColor@DetectPdfVersion
478           \ifnum\HyColor@PdfVersion<7 %
479             \HyColor@IfModel{#1}{%
480               \HyColor@IfXcolor{%
481                 \convertcolorspec\HyColor@model\HyColor@values
482                   \HyColor@model@rgb#4%
483                 \expandafter\HyColor@NormalizeCommaRGB#4\@nil#4%
484                 \edef#4{/C[#4]}% hash-ok
485               }{%
486                 \let#4\@empty
487                 \HyColor@ErrorModelNoXcolor{#5}{#6}%
488               }%
489             }{%
490               \HyColor@IfXcolor{%
491                 \extractcolorspec{#1}#4%
492                 \expandafter\convertcolorspec#4%
493                 \HyColor@model@rgb#4%
494                 \expandafter\HyColor@NormalizeCommaRGB#4\@nil#4%
495                 \edef#4{/C[#4]}% hash-ok
496               }{%
497                 \let#4\@empty
498                 \HyColor@ErrorSpecNoXcolor{#5}{#6}%
499               }%
500             }%
501           \else
502             \edef#4{/C[#3]}% hash-ok
503           \fi
504           \edef#3##1{%
```

```

505         #3 %
506         \noexpand\csname atfi@SETCMYKCOLOR##1\noexpand\endcsname
507     }%
508     \else
509         \ifx\HyColor@model\HyColor@model@empty
510             \PackageError{#5}{%
511                 Color model 'empty' is not permitted for option '#6'%
512             }\@ehc
513             \let#2\@empty
514             \let#3\@gobble
515             \let#4\@empty
516         \else
517             \ifx\HyColor@model\relax % (missing xcolor)
518                 \let#3\@gobble
519                 \let#4\@empty
520             \else
521                 \PackageError{#5}{%
522                     Internal error: unsupported color model%
523                 }\@ehc
524             \fi
525         \fi
526     \fi
527 \fi
528 \fi
529 \fi
530 }
531 </package>

```

### 3 Test

```

532 (*test1)
533 \ProvidesFile{hycolor-test1.tex}[2008/07/29 test file 1]
534 </test1>

535 (*test2)
536 \ProvidesFile{hycolor-test2.tex}[2008/07/29 test file 2]
537 \let\pdfmatch\relax
538 </test2>

539 (test3)\ProvidesFile{hycolor-test3.tex}[2008/07/29 test file 3]

540 (*test)

541 \documentclass{article}
542
543 \usepackage{qstest}
544 \IncludeTests{*}
545 \LogTests{log}{*}{*}
546
547 \makeatletter
548
549 \newcommand*{\TestPackageName}{test-package}
550 \newcommand*{\TestOptionName}{test-option}
551
552 \newcommand\Message{}
553 \def\Message#1#{\immediate\write16}
554
555 \newcommand*{\ExpectError}[2]{%
556     \begingroup
557         \global\let\saved@errhelp\errhelp
558         \global\let\saved@errmessage\errmessage
559         \let\errhelp\@gobble
560         \def\errmessage##1{%

```



```

561     \xdef\@ExpectErrorMessage{##1}%
562 }%
563 \PackageError\TestPackageName{#1}\@ehc
564 \def\errhelp##1{%
565     \global\let\errhelp\saved@errhelp
566 }%
567 \global\let\@ResultErrorMessage\@empty
568 \def\errmessage##1{%
569     \xdef\@ResultErrorMessage{##1}%
570     \global\let\errmessage\saved@errmessage
571     % \Message{[ ##1}%
572     % \Message{[] (ignored error)}%
573     % \Message{}%
574 }%
575 #2%
576 \endgroup
577 \Expect*{\@ResultErrorMessage}*{\@ExpectErrorMessage}%
578 }
579
580 \usepackage{hycolor}[2008/07/29]
581 /test)
582 (*test1)
583 \begin{qstest}{NumNormalize}{num, normalize}
584 \def\test#1#2{%
585     \HyColor@NormalizeNum{#1}\cmd
586     \Expect*{\cmd}{#2}%
587 }%
588 \test{0}{0}%
589 \test{000}{0}%
590 \test{-1}{0}%
591 \test{ 0 }{0}%
592 \test{1.1}{1}%
593 \test{100}{1}%
594 \test{00100}{1}%
595 \test{99.99}{1}%
596 \test{0.0}{0}%
597 \test{00.00}{0}%
598 \test{0.}{0}%
599 \test{.0}{0}%
600 \test{0.1}{.1}%
601 \test{0.10}{.1}%
602 \test{0.1000}{.1}%
603 \test{.1000}{.1}%
604 \test{0.01}{.01}%
605 \test{0.01010}{.0101}%
606 \test{.0000000001}{.0000000001}%
607 \test{.9999999999}{.9999999999}%
608 \end{qstest}
609
610 \begin{qstest}{BookmarkColor without xcolor}{bookmark, noxcolor}
611 \def\test#1#2{%
612     \HyColor@BookmarkColor{#1}\cmd\TestPackageName\TestOptionName
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 \def\errortest[#1]#2{%
620     \ExpectError{%
621         Color model ‘#1’ is not supported\MessageBreak
622         without package ‘xcolor’ in\MessageBreak

```

```

623     '\TestOptionName=[#1]{#2}'% hash-ok
624   }{%
625     \test{[#1]{#2}}{% hash-ok
626   }%
627 }%
628 \errortest[cmyk]{1,0,0,0}%
629 \errortest[empty]{}%
630 \def\errortest#1{%
631   \ExpectError{%
632     This color specification is not supported\MessageBreak
633     without package 'xcolor' in\MessageBreak
634     '\TestOptionName=#1'%
635   }{%
636     \test{#1}{}%
637   }%
638 }%
639 \end{qstest}
640 </test1>

641 (*test1 | test2)
642 \begin{qstest}{X0134 without xcolor}{X0134, noxcolor}
643   \def\test#1#2{%
644     \HyColor@XZeroOneThreeFour{#1}\cmd\TestPackageName\TestOptionName
645     \Expect*{\cmd}{#2}%
646   }%
647   \test{[empty]}{}{%
648     \test{[rgb]{1,0,0}}{1 0 0}%
649     \test{[gray]{0.10}}{.1}%
650     \test{[cmyk]{0,1,0,0}}{0 1 0 0}%
651     \test{[rgb]{ 1 , 1 , 0 }}{1 1 0}%
652     \def\errortest[#1]#2{%
653       \ExpectError{%
654         Color model '#1' is not supported\MessageBreak
655         without package 'xcolor' in\MessageBreak
656         'test-option=[#1]{#2}'% hash-ok
657       }{%
658         \HyColor@XZeroOneThreeFour{[#1]{#2}}\cmd
659         \TestPackageName\TestOptionName
660         \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
661       }%
662     }%
663     \errortest[Gray]{10}%
664     \errortest[cmy]{1,0,0}%
665     \def\errortest#1{%
666       \ExpectError{%
667         This color specification is not supported\MessageBreak
668         without package 'xcolor' in\MessageBreak
669         'test-option=#1'%
670       }{%
671         \HyColor@XZeroOneThreeFour{#1}\cmd\TestPackageName\TestOptionName
672         \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
673       }%
674     }%
675     \errortest{yellow}%
676   \end{qstest}
677
678 \begin{qstest}{HyperrefBorderColor without xcolor}%
679   {hyperef bordercolor, noxcolor}%
680   \def\test#1#2{%
681     \HyColor@HyperrefBorderColor{#1}\cmd\TestPackageName\TestOptionName
682     \Expect*{\cmd}{#2}%
683   }%
684   \test{[rgb]{1,0,0}}{1 0 0}%

```

```

685 \test{[gray]{0.10}}{.1 .1 .1}%
686 \test{[rgb]{ 1 , 1 , 0 }}{1 1 0}%
687 \def\errortest[#1]#2{%
688   \ExpectError{%
689     Color model ‘#1’ is not supported\MessageBreak
690     without package ‘xcolor’ in\MessageBreak
691     ‘test-option=[#1]{#2}’% hash-ok
692   }{%
693     \HyColor@HyperrefBorderColor{[#1]}{#2}}\cmd
694     \TestPackageName\TestOptionName
695     \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
696   }%
697 }%
698 \errortest[Gray]{10}%
699 \errortest[cmY]{1,0,0}%
700 \errortest[cmYk]{0,1,0,0}%
701 \def\errortest#1{%
702   \ExpectError{%
703     This color specification is not supported\MessageBreak
704     without package ‘xcolor’ in\MessageBreak
705     ‘test-option=#1’%
706   }{%
707     \HyColor@HyperrefBorderColor{#1}\cmd
708     \TestPackageName\TestOptionName
709     \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
710   }%
711 }%
712 \errortest{yellow}%
713 \end{qstest}
714 </test1 | test2>

715 <*test1 | test2>
716 \usepackage{xcolor}
717 \definecolor[named]{MyGreen}{rgb}{0,0.7,0}
718 \definecolor{mygreen}{named}{MyGreen}
719 </test1 | test2>

720 <*test1>
721 \begin{qstest}{BookmarkColor with xcolor}{bookmark, xcolor}
722   \def\test#1#2{%
723     \HyColor@BookmarkColor{#1}\cmd\PackageName\OptionName
724     \Expect*{\cmd}{#2}%
725   }%
726   \test{[rgb]{1,0,0}}{1 0 0}%
727   \test{[gray]{0.10}}{.1 .1 .1}%
728   \test{}{}%
729   \test{[rgb]{ 1 , 1 , 0 }}{1 1 0}%
730   \test{[cmYk]{1,0,0,0}}{0 1 1}%
731   \test{red}{1 0 0}%
732   \test{cyan}{0 1 1}%
733   \test{red!40!blue}{.4 0 .6}%
734   \test{[Gray]{10}}{.66667 .66667 .66667}%
735   \test{[RGB]{100,200,50}}{.39217 .78432 .19609}%
736   \test{[wave]{363}}{.00316 0 .00316}%
737   \test{[wave]814}{.00797 0 0}%
738   \test{[HSB]{100,200,50}}{.03473 .20833 .12152}%
739   \test{[HTML]{A800FF}}{.65881 0 1}%
740   \test{[cmY]{.3,.5,.2}}{.7 .5 .8}%
741   \test{[cmYk]{.3,.5,.2,.1}}{.6 .4 .7}%
742   \test{[hsb]{.3,.5,.2}}{.12 .2 .1}%
743   \test{[Hsb]{120,.5,.2}}{.1 .2 .1}%
744   \test{[tHsb]{120,.5,.2}}{.2 .2 .1}%
745   \test{[named]{MyGreen}}{0 .7 0}%
746   \test{mygreen}{0 .7 0}%

```

```

747 \end{qstest}
748
749 \begin{qstest}{HyperrefColor}{hyperref, color}
750   \def\test#1#2{%
751     \HyColor@HyperrefColor{#1}\cmd
752     \Expect*{\cmd}{#2}%
753   }%
754   \test{red}{red}%
755   \test{[rgb]{1,0,0}}{[rgb]{1,0,0}}%
756   \HyColor@HyperrefColor{}\cmd
757   \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
758 \end{qstest}
759 </test1>

760 <*test1 | test2>
761 \begin{qstest}{X0134 with xcolor}{hyperref, X0134, xcolor}
762   \def\test#1#2{%
763     \HyColor@XZeroOneThreeFour{#1}\cmd\PackageName\OptionName
764     \Expect*{\cmd}{#2}%
765   }%
766   \test{[empty]{}{}}{%
767     \test{[gray]{0.1}}{.1}%
768     \test{[rgb]{1,0.5,0.0}}{1 .5 0}%
769     \test{[cmyk]{0,1,0,0.5}}{0 1 0 .5}%
770     \test{[Gray]{10}}{.66667}%
771     \test{red}{1 0 0}%
772     \test{1 0 0}{1 0 0}%
773     \test{001.0 .23 0}{1 .23 0}%
774     \test{[named]{MyGreen}}{0 .7 0}%
775     \test{mygreen}{0 .7 0}%
776     \HyColor@XZeroOneThreeFour{}\cmd\PackageName\OptionName
777     \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
778 \end{qstest}
779
780 \begin{qstest}{FieldColor}{hyperref, field, FieldColor}
781   \def\test#1#2{%
782     \HyColor@FieldColor{#1}\cmd\PackageName\OptionName
783     \Expect*{\cmd}{#2}%
784   }%
785   \test{}{}%
786   \test{[gray]{0.7}}{.7 g}%
787   \test{[rgb]{1,0,0}}{1 0 0 rg}%
788   \test{[cmyk]{0,1,0,0}}{0 1 0 0 k}%
789   \test{[cmy]{.5,.4,.3}}{.5 .6 .7 rg}%
790 \end{qstest}
791 </test1 | test2>

```

### 3.1 Test for package attachfile2

```

792 <*test3>
793 \def\atfi@SETRGBCOLORTest{set-rgb}
794 \def\atfi@SETGRAYCOLORTest{set-gray}
795 \def\atfi@SETCMYKCOLORTest{set-cmyk}
796 \def\Test#1#2#3#4#5{%
797   \begingroup
798     \setbox0=\hbox{%
799       \begingroup
800         \chardef\HyColor@PdfVersion=6 %
801         \HyColor@AttachfileColor{#1}\spec\inlinemacro\annot
802         \TestPackageName\TestOptionName
803         \edef\inline{\inlinemacro{test}}%
804         \expandafter\Expect\expandafter{\spec}{#2}%
805         \expandafter\Expect\expandafter{\inline}{#3}%
806         \expandafter\Expect\expandafter{\annot}{#4}%

```

```

807 \endgroup
808 \begingroup
809 \chardef\HyColor@PdfVersion=7 %
810 \HyColor@AttachfileColor{#1}\spec\inlinemacro\annot
811 \TestPackageName\TestOptionName
812 \edef\inline{\inlinemacro{test}}%
813 \expandafter\Expect\expandafter{\spec}{#2}%
814 \expandafter\Expect\expandafter{\inline}{#3}%
815 \expandafter\Expect\expandafter{\annot}{#5}%
816 \endgroup
817 }%
818 \Expect*{\the\wd0}{0.0pt}%
819 \endgroup
820 }
821 \newif\ifError
822 \def\TestError[#1]#2#3#4#5#6{%
823 \begingroup
824 \global\Errorfalse
825 \let\OrgPackageError\PackageError
826 \def\PackageError##1##2##3{%
827 \edef\TestTemp{##1}%
828 \ifx\TestTemp\TestPackageName
829 \Expect*{\ifError too many errors\else ok\fi}{ok}%
830 \Expect*{#6}*{##2}%
831 \global\Errortrue
832 \else
833 \OrgPackageError{##1}{##2}{##3}%
834 \fi
835 }%
836 \setbox0=\hbox{%
837 \begingroup
838 \chardef\HyColor@PdfVersion=#1 %
839 \HyColor@AttachfileColor{#2}\spec\inlinemacro\annot
840 \TestPackageName\TestOptionName
841 \edef\inline{\inlinemacro{test}}%
842 \expandafter\Expect\expandafter{\spec}{#3}%
843 \expandafter\Expect\expandafter{\inline}{#4}%
844 \expandafter\Expect\expandafter{\annot}{#5}%
845 \endgroup
846 \ifx\#6\%
847 \else
848 \Expect*{\ifError ok\else missing error\fi}{ok}%
849 \fi
850 }%
851 \Expect*{\the\wd0}{0.0pt}%
852 \endgroup
853 }
854 \def\NoEmptyModel{%
855 Color model 'empty' is not permitted for option '\TestOptionName'%
856 }
857 \def\ModelNoXcolor#1#2{%
858 Color model '#1' is not supported\MessageBreak
859 without package 'xcolor' in\MessageBreak
860 '\TestOptionName=[#1]{#2}'% hash-ok
861 }
862 \def\SpecNoXColor#1{%
863 This color specification is not supported\MessageBreak
864 without package 'xcolor' in\MessageBreak
865 'test-option=#1'%
866 }
867 \begin{qstest}{AttachfileColor}{AttachfileColor}
868 \Test{}{}{}{}{}%

```



## 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  $\text{\TeX}$  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- $\text{\TeX}$ :

```
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>test/hycolor-test3.tex</code>	$\rightarrow$ <code>doc/latex/oberdiek/test/hycolor-test3.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  $\text{\TeX}$  distribution (te $\text{\TeX}$ , mi $\text{\TeX}$ , ...) relies on file name databases, you must refresh these. For example, te $\text{\TeX}$  users run `texhash` or `mktextlsr`.

---

<sup>1</sup>[ftp://ftp.ctan.org/tex-archive/](http://ftp.ctan.org/tex-archive/)

## 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 `pdf $\LaTeX$` :

```
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.

## 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>	.....	364	
<code>\!</code>	.....	6	<code>\:</code> ..... 7
<code>\"</code>	.....	11	<code>\;</code> ..... 10
<code>\+</code>	.....	9	<code>\&gt;</code> ..... 12
<code>\-</code>	.....	8	<code>\@ExpectErrorMessage</code> ..... 561, 577





\HyColor@NormalizeCommaRGB	77, 100, 103, 122, 217, 234, 247, 257, 271, 388, 391, 409, 414, 483, 494
\HyColor@NormalizeNum	..... 35, 78, 80, 82, 86, 88, 90, 92, 106, 214, 224, 254, 265, 394, 585
\HyColor@one	..... 342, 352
\HyColor@PdfVersion	..... 426, 428, 466, 478, 800, 809, 838
\HyColor@resultfalse	..... 331, 347, 356, 378, 450
\HyColor@resulttrue	..... 320, 375
\HyColor@ReverseString	48, 53, 60, 65
\HyColor@SpaceToComma	..... 432, 454
\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, 386, 391, 395, 404, 408, 409, 481
\HyColor@WithModel	..... 146, 152
\HyColor@WithoutModel	..... 148, 157
\HyColor@XZeroOneThreeFour	..... 209, 282, 285, 451, 644, 658, 671, 763, 776
\HyColor@zero	..... 341, 350
<b>I</b>	
\ifdim	..... 36
\ifError	..... 821, 829, 848
\ifHyColor@result	..... 303, 324, 326, 333, 452
\ifnum	..... 44, 367, 466, 478
\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, 305, 322, 346, 350, 352, 354, 390, 393, 405, 425, 434, 446, 457, 464, 476, 509, 517, 660, 672, 695, 709, 757, 777, 828, 846
\immediate	..... 553
\IncludeTests	..... 544
\inline	... 803, 805, 812, 814, 841, 843
\inlinemacro	801, 803, 810, 812, 839, 841
<b>L</b>	
\LogTests	..... 545
<b>M</b>	
\makeatletter	..... 547
\Message	..... 552, 553, 571, 572, 573
\MessageBreak	..... 132, 133, 139, 140, 621, 622, 632, 633, 654, 655, 667, 668, 689, 690, 703, 704, 858, 859, 863, 864
\ModelNoXcolor	..... 857, 882
<b>N</b>	
\NeedsTeXFormat	..... 2
\newcommand	..... 549, 550, 552, 555
\newif	..... 303, 821
\NoEmptyModel	..... 854, 878, 879, 916, 917
<b>O</b>	
\OptionName	..... 723, 763, 776, 782
\OrgPackageError	..... 825, 833
<b>P</b>	
\PackageError	..... 131, 138, 298, 510, 521, 563, 825, 826
\PackageName	..... 723, 763, 776, 782
\pdflastmatch	..... 371, 372, 373
\pdfmatch	..... 367, 537
\ProvidesFile	..... 533, 536, 539
\ProvidesPackage	..... 3
<b>S</b>	
\saved@errhelp	..... 557, 565
\saved@errmessage	..... 558, 570
\setbox	..... 798, 836
\space	..... 318, 367, 368
\spec	..... 801, 804, 810, 813, 839, 842
\SpecNoXColor	..... 862, 885, 886
\strip@prefix	..... 371, 372, 373
<b>T</b>	
\Test	..... 796, 868, 869, 871, 873, 875, 877, 894, 895, 897, 899, 901, 903, 904, 905, 906, 907, 909, 910, 912, 913, 914
\test	584, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 611, 615, 616, 617, 618, 625, 636, 643, 647, 648, 649, 650, 651, 680, 684, 685, 686, 722, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 750, 754, 755, 762, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 781, 785, 786, 787, 788, 789
\TestError	..... 822, 878, 879, 880, 883, 885, 886, 916, 917
\TestOptionName	..... 550, 612, 623, 634, 644, 659, 671, 681, 694, 708, 802, 811, 840, 855, 860
\TestPackageName	..... 549, 563, 612, 644, 659, 671, 681, 694, 708, 802, 811, 828, 840
\TestTemp	..... 827, 828
\the	..... 818, 851
<b>U</b>	
\usepackage	..... 543, 580, 716, 888

W		Y	
\wd	818, 851	\y	22, 34
\write	553		
X		Z	
\x	13, 27, 30, 34, 312, 317	\z@	36, 368
		\zap@space	39