

# The hycolor package

Heiko Oberdiek  
<heiko.oberdiek at googlemail.com>

2009/12/12 v1.6

## 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	10
2.5.3	Field options	12
2.5.4	Detection for naked RGB values	12
2.5.5	Options *bordercolor	14
2.6	Package attachfile2	15
2.7	Patch for package xcolor	17
2.7.1	Fix fragile \@frameb@x	20
<b>3</b>	<b>Test</b>	<b>20</b>
3.1	Test for package attachfile2	25
3.2	Test for package xcolor	27
3.2.1	Test for \@frameb@x/\fbox	28
<b>4</b>	<b>Installation</b>	<b>28</b>
4.1	Download	28
4.2	Bundle installation	28
4.3	Package installation	28
4.4	Refresh file name databases	29
4.5	Some details for the interested	29
<b>5</b>	<b>History</b>	<b>29</b>
	[2007/04/09 v1.0]	29
	[2007/04/11 v1.1]	30
	[2008/07/29 v1.2]	30
	[2008/08/01 v1.3]	30
	[2008/09/08 v1.4]	30
	[2009/10/02 v1.5]	30
	[2009/12/12 v1.6]	30

## 1 Documentation

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

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

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

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

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

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

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

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

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

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

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

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

### 1.1 Summary

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

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

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

## 2 Implementation

```

1 <*package>
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{hycolor}%
4 [2009/12/12 v1.6 Color options of hyperref/bookmark (H0)]%
5 \RequirePackage{xcolor-patch}[2009/12/12]

```

### 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;
  endsw
  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
endsw
otherwise
  | if with xcolor then
    | (model, values ← get model and values;
    | cmd ← ConvertToRGB(model, values);
  else
    | error;
  end
endsw
endsw

```

---

```

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;
  endsw
  case with model
    | Call \color with model;
  endsw
  case without model
    | Call \color without model;
  endsw
endsw

```

---

```

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

```

---

\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\curname pdfmatch\endcsname\relax
307 \expandafter\@firstoftwo
308 \else
309     \expandafter\@secondoftwo
310 \fi
311 {%
312     \begingroup
313     \def\x#1{\endgroup
314         \def\HyColor@IfRGB##1{%
315             \HyColor@@IfRGB##1#1#1\@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 ← no color;
  endsw
  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;
        endsw
        else
          | error(package, option, "Missing xcolor");
          | cmd ← no color;
        end
      endsw
    end
  endsw
  case rgb values
    | cmd ← output values;
  endsw
  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
  endsw
endsw

```

---

`\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\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}[2009/12/12 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@cnv@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

```

### 2.7.1 Fix fragile \@frameb@x

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

```

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

693 \ifundefined{XC@frameb@x }{%
694   \expandafter\let\csname XC@frameb@x \endcsname\XC@frameb@x
695   \edef\XC@frameb@x{%
696     \noexpand\protect
697     \expandafter\noexpand\csname XC@frameb@x \endcsname
698   }%
699   \expandafter\ifx\csname XC@frameb@x \endcsname\@frameb@x
700     \let\@frameb@x\XC@frameb@x
701   \fi
702 }{%
703 }
704 \</xcolor>

```

## 3 Test

```

705 <*test1>
706 \ProvidesFile{hycolor-test1.tex}[2009/12/12 test file 1]
707 </test1>

708 <*test2>
709 \ProvidesFile{hycolor-test2.tex}[2009/12/12 test file 2]
710 \let\pdfmatch\relax
711 </test2>

712 <test3>\ProvidesFile{hycolor-test3.tex}[2009/12/12 test file 3]
713 <*test>

714 \documentclass{article}
715

```

```

716 \usepackage{qstest}
717 \IncludeTests{*}
718 \LogTests{log}{*}{*}
719
720 \makeatletter
721
722 \newcommand*{\TestPackageName}{test-package}
723 \newcommand*{\TestOptionName}{test-option}
724
725 \newcommand\ErrorMessage{}
726 \def\Message#1#{\immediate\write16}
727
728 \newcommand*{\ExpectError}[2]{%
729   \begingroup
730     \global\let\saved@errhelp\errhelp
731     \global\let\saved@errmessage\errmessage
732     \let\errhelp@gobble
733     \def\errmessage##1{%
734       \xdef\@ExpectErrorMessage{##1}%
735     }%
736     \PackageError\TestPackageName{#1}\@ehc
737     \def\errhelp##1{%
738       \global\let\errhelp\saved@errhelp
739     }%
740     \global\let\@ResultErrorMessage\@empty
741     \def\errmessage##1{%
742       \xdef\@ResultErrorMessage{##1}%
743       \global\let\errmessage\saved@errmessage
744       % \Message{[ #1]}%
745       % \Message{ (ignored error)}%
746       % \Message{}%
747     }%
748     #2%
749   \endgroup
750   \Expect*{\@ResultErrorMessage}*{\@ExpectErrorMessage}%
751 }
752 \usepackage{scrfile}
753 \usepackage{hycolor}[2009/12/12]
754 \</test>
755 <*test1>
756 \begin{qstest}{NumNormalize}{num, normalize}
757   \def\test#1#2{%
758     \HyColor@NormalizeNum{#1}\cmd
759     \Expect*{\cmd}{#2}%
760   }%
761   \test{0}{0}%
762   \test{000}{0}%
763   \test{-1}{0}%
764   \test{ 0 }{0}%
765   \test{1.1}{1}%
766   \test{100}{1}%
767   \test{00100}{1}%
768   \test{99.99}{1}%
769   \test{0.0}{0}%
770   \test{00.00}{0}%
771   \test{0.}{0}%
772   \test{.0}{0}%
773   \test{0.1}{.1}%
774   \test{0.10}{.1}%
775   \test{0.1000}{.1}%
776   \test{.1000}{.1}%
777   \test{0.01}{.01}%

```

```

778 \test{0.01010}{.0101}%
779 \test{.0000000001}{.0000000001}%
780 \test{.9999999999}{.9999999999}%
781 \end{qstest}
782
783 \begin{qstest}{BookmarkColor without xcolor}{bookmark, noxcolor}
784 \def\test#1#2{%
785   \HyColor@BookmarkColor{#1}\cmd\TestPackageName\TestOptionName
786   \Expect*{\cmd}{#2}%
787 }%
788 \test{[rgb]{1,0,0}}{1 0 0}%
789 \test{[gray]{0.10}}{.1 .1 .1}%
790 \test{}{}%
791 \test{[rgb]{ 1 , 1 , 0 }}{1 1 0}%
792 \def\errortest[#1]#2{%
793   \ExpectError{%
794     Color model '#1' is not supported\MessageBreak
795     without package 'xcolor' in\MessageBreak
796     '\TestOptionName=[#1]{#2}'% hash-ok
797   }{%
798     \test{[#1]{#2}}{}% hash-ok
799   }%
800 }%
801 \errortest[cmk]{1,0,0,0}%
802 \errortest[empty]{}%
803 \def\errortest#1{%
804   \ExpectError{%
805     This color specification is not supported\MessageBreak
806     without package 'xcolor' in\MessageBreak
807     '\TestOptionName=#1'%
808   }{%
809     \test{#1}{}%
810   }%
811 }%
812 \end{qstest}
813 \test1

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

```

```

840     This color specification is not supported\MessageBreak
841     without package 'xcolor' in\MessageBreak
842     'test-option=#1'%
843   }{%
844     \HyColor@XZeroOneThreeFour{#1}\cmd\TestPackageName\TestOptionName
845     \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
846   }%
847 }%
848 \errortest{yellow}%
849 \end{qstest}
850
851 \begin{qstest}{\HyperrefBorderColor without xcolor}%
852     {hyperref bordercolor, noxcolor}%
853 \def\test#1#2{%
854     \HyColor@HyperrefBorderColor{#1}\cmd\TestPackageName\TestOptionName
855     \Expect*{\cmd}{#2}%
856 }%
857 \test{[rgb]{1,0,0}}{1 0 0}%
858 \test{[gray]{0.10}}{.1 .1 .1}%
859 \test{[rgb]{ 1 , 1 , 0 }}{1 1 0}%
860 \def\errortest[#1]#2{%
861     \ExpectError{%
862         Color model '#1' is not supported\MessageBreak
863         without package 'xcolor' in\MessageBreak
864         'test-option=[#1]{#2}'% hash-ok
865     }{%
866         \HyColor@HyperrefBorderColor{[#1]{#2}}\cmd
867         \TestPackageName\TestOptionName
868         \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
869     }%
870 }%
871 \errortest[Gray]{10}%
872 \errortest[cmY]{1,0,0}%
873 \errortest[cmYk]{0,1,0,0}%
874 \def\errortest#1{%
875     \ExpectError{%
876         This color specification is not supported\MessageBreak
877         without package 'xcolor' in\MessageBreak
878         'test-option=#1'%
879     }{%
880         \HyColor@HyperrefBorderColor{#1}\cmd
881         \TestPackageName\TestOptionName
882         \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
883     }%
884 }%
885 \errortest{yellow}%
886 \end{qstest}
887 </test1 | test2>

888 <*test1 | test2>
889 \usepackage{xcolor}
890 \definecolor[named]{MyGreen}{rgb}{0,0.7,0}
891 \definecolor{mygreen}{named}{MyGreen}
892 </test1 | test2>

893 <*test1>
894 \begin{qstest}{BookmarkColor with xcolor}{bookmark, xcolor}
895 \def\test#1#2{%
896     \HyColor@BookmarkColor{#1}\cmd\PackageName\OptionName
897     \Expect*{\cmd}{#2}%
898 }%
899 \test{[rgb]{1,0,0}}{1 0 0}%
900 \test{[gray]{0.10}}{.1 .1 .1}%
901 \test{}{}%

```

```

902 \test{[rgb]{ 1 , 1 , 0 }}{1 1 0}%
903 \test{[cmyk]{1,0,0,0}}{0 1 1}%
904 \test{red}{1 0 0}%
905 \test{cyan}{0 1 1}%
906 \test{red!40!blue}{.4 0 .6}%
907 \test{[Gray]{10}}{.66667 .66667 .66667}%
908 \test{[RGB]{100,200,50}}{.39217 .78432 .19609}%
909 \test{[wave]{363}}{.00316 0 .00316}%
910 \test{[wave]814}{.00797 0 0}%
911 \test{[HSB]{100,200,50}}{.03473 .20833 .12152}%
912 \test{[HTML]{A800FF}}{.65881 0 1}%
913 \test{[cmy]{.3,.5,.2}}{.7 .5 .8}%
914 \test{[cmyk]{.3,.5,.2,.1}}{.6 .4 .7}%
915 \test{[hsb]{.3,.5,.2}}{.12 .2 .1}%
916 \test{[Hsb]{120,.5,.2}}{.1 .2 .1}%
917 \test{[tHsb]{120,.5,.2}}{.2 .2 .1}%
918 \test{[named]{MyGreen}}{0 .7 0}%
919 \test{mygreen}{0 .7 0}%
920 \end{qstest}
921
922 \begin{qstest}{HyperrefColor}{hyperref, color}
923 \def\test#1#2{%
924 \HyColor@HyperrefColor{#1}\cmd
925 \Expect*{\cmd}{#2}%
926 }%
927 \test{red}{red}%
928 \test{[rgb]{1,0,0}}{[rgb]{1,0,0}}%
929 \HyColor@HyperrefColor{}\cmd
930 \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
931 \end{qstest}
932 \test1)

933 (*test1 | test2)
934 \begin{qstest}{X0134 with xcolor}{hyperref, X0134, xcolor}
935 \def\test#1#2{%
936 \HyColor@XZeroOneThreeFour{#1}\cmd\PackageName\OptionName
937 \Expect*{\cmd}{#2}%
938 }%
939 \test{[empty]{}{}}{%
940 \test{[gray]{0.1}}{.1}%
941 \test{[rgb]{1,0.5,0.0}}{1 .5 0}%
942 \test{[cmyk]{0,1,0,0.5}}{0 1 0 .5}%
943 \test{[Gray]{10}}{.66667}%
944 \test{red}{1 0 0}%
945 \test{1 0 0}{1 0 0}%
946 \test{001.0 .23 0}{1 .23 0}%
947 \test{[named]{MyGreen}}{0 .7 0}%
948 \test{mygreen}{0 .7 0}%
949 \HyColor@XZeroOneThreeFour{}\cmd\PackageName\OptionName
950 \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
951 \end{qstest}
952
953 \begin{qstest}{FieldColor}{hyperref, field, FieldColor}
954 \def\test#1#2{%
955 \HyColor@FieldColor{#1}\cmd\PackageName\OptionName
956 \Expect*{\cmd}{#2}%
957 }%
958 \test{}{ }%
959 \test{[gray]{0.7}}{.7 g}%
960 \test{[rgb]{1,0,0}}{1 0 0 rg}%
961 \test{[cmyk]{0,1,0,0}}{0 1 0 0 k}%
962 \test{[cmy]{.5,.4,.3}}{.5 .6 .7 rg}%
963 \end{qstest}

```



964 </test1 | test2>

### 3.1 Test for package attachfile2

```
965 <*test3>
966 \def\atfi@SETRGBCOLORtest{set-rgb}
967 \def\atfi@SETGRAYCOLORtest{set-gray}
968 \def\atfi@SETCMYKCOLORtest{set-cmyk}
969 \def\Test#1#2#3#4#5{%
970   \begingroup
971     \setbox0=\hbox{%
972       \begingroup
973         \chardef\HyColor@PdfVersion=6 %
974         \HyColor@AttachfileColor{#1}\spec\inlinemacro\annot
975           \TestPackageName\TestOptionName
976         \edef\inline{\inlinemacro{test}}%
977         \expandafter\Expect\expandafter{\spec}{#2}%
978         \expandafter\Expect\expandafter{\inline}{#3}%
979         \expandafter\Expect\expandafter{\annot}{#4}%
980       \endgroup
981       \begingroup
982         \chardef\HyColor@PdfVersion=7 %
983         \HyColor@AttachfileColor{#1}\spec\inlinemacro\annot
984           \TestPackageName\TestOptionName
985         \edef\inline{\inlinemacro{test}}%
986         \expandafter\Expect\expandafter{\spec}{#2}%
987         \expandafter\Expect\expandafter{\inline}{#3}%
988         \expandafter\Expect\expandafter{\annot}{#5}%
989       \endgroup
990     }%
991     \Expect*{\the\wd0}{0.0pt}%
992   \endgroup
993 }
994 \newif\ifError
995 \def\TestError[#1]#2#3#4#5#6{%
996   \begingroup
997     \global\Errorfalse
998     \let\OrgPackageError\PackageError
999     \def\PackageError##1##2##3{%
1000       \edef\TestTemp{##1}%
1001       \ifx\TestTemp\TestPackageName
1002         \Expect*{\ifError too many errors\else ok\fi}{ok}%
1003         \Expect*{#6}*{##2}%
1004         \global\Errortrue
1005       \else
1006         \OrgPackageError{##1}{##2}{##3}%
1007       \fi
1008     }%
1009     \setbox0=\hbox{%
1010       \begingroup
1011         \chardef\HyColor@PdfVersion=#1 %
1012         \HyColor@AttachfileColor{#2}\spec\inlinemacro\annot
1013           \TestPackageName\TestOptionName
1014         \edef\inline{\inlinemacro{test}}%
1015         \expandafter\Expect\expandafter{\spec}{#3}%
1016         \expandafter\Expect\expandafter{\inline}{#4}%
1017         \expandafter\Expect\expandafter{\annot}{#5}%
1018       \endgroup
1019       \ifx\#6\%
1020       \else
1021         \Expect*{\ifError ok\else missing error\fi}{ok}%
1022       \fi
1023     }%
```

```

1024 \Expect*{\the\wd0}{0.0pt}%
1025 \endgroup
1026 }
1027 \def\NoEmptyModel{%
1028 Color model 'empty' is not permitted for option '\TestOptionName'%
1029 }
1030 \def\ModelNoXcolor#1#2{%
1031 Color model '#1' is not supported\MessageBreak
1032 without package 'xcolor' in\MessageBreak
1033 '\TestOptionName=[#1]{#2}'% hash-ok
1034 }
1035 \def\SpecNoXColor#1{%
1036 This color specification is not supported\MessageBreak
1037 without package 'xcolor' in\MessageBreak
1038 'test-option=#1'%
1039 }
1040 \begin{qstest}{AttachfileColor}{AttachfileColor}
1041 \Test{}{}{}{}{}%
1042 \Test{0.1 0.2 0.3}{[rgb]{.1,.2,.3}}{.1 .2 .3 set-rgb}%
1043 {/C[.1 .2 .3]}/{C[.1 .2 .3]}%
1044 \Test{[gray]{0.4}}{[gray]{0.4}}{.4 set-gray}%
1045 {/C[.4 .4 .4]}/{C[.4]}%
1046 \Test{[rgb]{0.3,.2,.1}}{[rgb]{0.3,.2,.1}}{.3 .2 .1 set-rgb}%
1047 {/C[.3 .2 .1]}/{C[.3 .2 .1]}%
1048 \Test{0.0 1.0 1}{[rgb]{0,1,1}}{0 1 1 set-rgb}%
1049 {/C[0 1 1]}/{C[0 1 1]}%
1050 \Test{[gray]1}{[gray]1}{1 set-gray}{/C[1 1 1]}/{C[1]}%
1051 \TestError[6]{[empty]{}{}{}{}{}{}\NoEmptyModel
1052 \TestError[7]{[empty]{}{}{}{}{}{}\NoEmptyModel
1053 \TestError[6]{[cmyk]{.1,.2,.3,.4}}{[cmyk]{.1,.2,.3,.4}}%
1054 {.1 .2 .3 .4 set-cmyk}{}%
1055 {\ModelNoXcolor{cmyk}{.1,.2,.3,.4}}%
1056 \TestError[7]{[cmyk]{.1,.2,.3,.4}}{[cmyk]{.1,.2,.3,.4}}%
1057 {.1 .2 .3 .4 set-cmyk}{/C[.1 .2 .3 .4]}{}%
1058 \TestError[6]{red}{red}{}{}{}\SpecNoXColor{red}}%
1059 \TestError[7]{red}{red}{}{}{}\SpecNoXColor{red}}%
1060 \end{qstest}
1061 \usepackage{xcolor}
1062 \definecolor[named]{MyGreen}{rgb}{0,0.7,0}
1063 \definecolor{mygreen}{named}{MyGreen}
1064 \definecolor{graynine}{gray}{0.9}
1065 \definecolor{GraySix}{Gray}{9}
1066 \begin{qstest}{AttachfileColorX}{AttachfileColorX}
1067 \Test{}{}{}{}{}%
1068 \Test{0.1 0.2 0.3}{[rgb]{.1,.2,.3}}{.1 .2 .3 set-rgb}%
1069 {/C[.1 .2 .3]}/{C[.1 .2 .3]}%
1070 \Test{[gray]{0.4}}{[gray]{0.4}}{.4 set-gray}%
1071 {/C[.4 .4 .4]}/{C[.4]}%
1072 \Test{[rgb]{0.3,.2,.1}}{[rgb]{0.3,.2,.1}}{.3 .2 .1 set-rgb}%
1073 {/C[.3 .2 .1]}/{C[.3 .2 .1]}%
1074 \Test{0.0 1.0 1}{[rgb]{0,1,1}}{0 1 1 set-rgb}%
1075 {/C[0 1 1]}/{C[0 1 1]}%
1076 \Test{[gray]1}{[gray]1}{1 set-gray}{/C[1 1 1]}/{C[1]}%
1077 \Test{red}{red}{1 0 0 set-rgb}{/C[1 0 0]}/{C[1 0 0]}%
1078 \Test{black}{black}{0 set-gray}{/C[0 0 0]}/{C[0]}%
1079 \Test{cyan}{cyan}{1 0 0 0 set-cmyk}{/C[0 1 1]}/{C[1 0 0 0]}%
1080 \Test{[named]{black}}{[named]{black}}{0 0 0 set-rgb}%
1081 {/C[0 0 0]}/{C[0 0 0]}%
1082 \Test{[Gray]{9}}{[Gray]{9}}{.6 set-gray}{/C[.6 .6 .6]}/{C[.6]}%
1083 \Test{[HTML]{0080FF}}{[HTML]{0080FF}}{0 .50195 1 set-rgb}%
1084 {/C[0 .50195 1]}/{C[0 .50195 1]}%
1085 \Test{graynine}{graynine}{.9 set-gray}{/C[.9 .9 .9]}/{C[.9]}%

```



```

1146 \xcol3\end{document}
1147 \xcol1 | xcol2\@@end
1148 \test-xcolor)

```

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

```

1149 \*test-xcolor-fbox)
1150 \NeedsTeXFormat{LaTeX2e}
1151 \documentclass{article}
1152 \usepackage{xcolor}
1153 \usepackage{xcolor-patch}[2009/12/12]
1154 \makeatletter
1155 \protected@edef\x{\fbox{abc}}
1156 \let\@tempa\@undefined
1157 \protected@edef\x{\fbox{abc}}
1158 \makeatother
1159 \begin{document}
1160 \MakeUppercase{\fbox{abc}}
1161 \end{document}
1162 \test-xcolor-fbox)

```

## 4 Installation

### 4.1 Download

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

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

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

**Bundle.** All the packages of the bundle ‘oberdiek’ are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

[CTAN:install/macros/latex/contrib/oberdiek.tds.zip](#)

*TDS* refers to the standard “A Directory Structure for  $\TeX$  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
```

---

<sup>1</sup><http://ftp.ctan.org/tex-archive/>

**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>	→ <code>tex/latex/oberdiek/hycolor.sty</code>
<code>xcolor-patch.sty</code>	→ <code>tex/latex/oberdiek/xcolor-patch.sty</code>
<code>hycolor.pdf</code>	→ <code>doc/latex/oberdiek/hycolor.pdf</code>
<code>test/hycolor-test1.tex</code>	→ <code>doc/latex/oberdiek/test/hycolor-test1.tex</code>
<code>test/hycolor-test2.tex</code>	→ <code>doc/latex/oberdiek/test/hycolor-test2.tex</code>
<code>test/hycolor-test3.tex</code>	→ <code>doc/latex/oberdiek/test/hycolor-test3.tex</code>
<code>test/hycolor-test-xcol1.tex</code>	→ <code>doc/latex/oberdiek/test/hycolor-test-xcol1.tex</code>
<code>test/hycolor-test-xcol2.tex</code>	→ <code>doc/latex/oberdiek/test/hycolor-test-xcol2.tex</code>
<code>test/hycolor-test-xcol3.tex</code>	→ <code>doc/latex/oberdiek/test/hycolor-test-xcol3.tex</code>
<code>test/hycolor-test-xcol4.tex</code>	→ <code>doc/latex/oberdiek/test/hycolor-test-xcol4.tex</code>
<code>hycolor.dtx</code>	→ <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 $\text{\TeX}$` , `mik $\text{\TeX}$` , ...) relies on file name databases, you must refresh these. For example, `te $\text{\TeX}$`  users run `texhash` or `mktextlsr`.

## 4.5 Some details for the interested

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

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

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

If you insist on using  $\text{\LaTeX}$  for `docstrip` (really, `docstrip` does not need  $\text{\LaTeX}$ ), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{hycolor.dtx}
```

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

**Generating the documentation.** You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by the configuration file `ltxdoc.cfg`. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with `pdf $\text{\LaTeX}$` :

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

# 5 History

[2007/04/09 v1.0]

- First version.

## [2007/04/11 v1.1]

- Line ends sanitized.

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

- Support for package `attachfile2` added.

## [2008/08/01 v1.3]

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

## [2008/09/08 v1.4]

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

## [2009/10/02 v1.5]

- Doku fixes (Herbert Voss).

## [2009/12/12 v1.6]

- Short info shortened.

## 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		290, 347, 406, 447, 449, 487, 498, 514, 516, 520, 740, 1133, 1138
<code>\!</code> .....	7	
<code>\"</code> .....	12	<code>\@expandtwoargs</code> ..... 584, 598
<code>\+</code> .....	10	<code>\@firstofone</code> ..... 439, 537
<code>\-</code> .....	9	<code>\@firstoftwo</code> 156, 169, 252, 307, 337, 377
<code>\.</code> .....	365	<code>\@for</code> ..... 1132, 1137
<code>\:</code> .....	8	<code>\@frameb@x</code> ..... 699, 700
<code>\;</code> .....	11	<code>\@gobble</code> ..... 436, 448, 515, 519, 732
<code>\&gt;</code> .....	13	<code>\@ifnextchar</code> ..... 146, 192
<code>\@end</code> .....	1147	<code>\@ifpackageloaded</code> ..... 536, 539, 546
<code>\@scl</code> .....	625,	<code>\@ifundefined</code> ..... 693
	630, 636, 641, 667, 672, 678, 683	<code>\@one</code> ..... 45
<code>\@tmp</code> .....	558,	<code>\@nil</code> ..... 41, 44, 49,
	563, 566, 583, 589, 592, 613,	54, 61, 66, 78, 86, 101, 104, 123,
	614, 617, 618, 621, 622, 626,	151, 153, 158, 187, 194, 197,
	627, 631, 632, 633, 637, 638,	218, 220, 235, 248, 259, 262,
	642, 646, 647, 654, 655, 659,	272, 315, 320, 350, 389, 392,
	660, 663, 664, 668, 669, 673,	410, 415, 433, 442, 455, 484, 495
	674, 675, 679, 680, 684, 688, 689	<code>\@nnil</code> ..... 566, 578, 592, 606
<code>\@ExpectErrorMessage</code> .....	734, 750	<code>\@onelevel@sanitize</code> .....
<code>\@ResultErrorMessage</code> ..	740, 742, 750	..... 32, 173, 175, 177, 179, 181
<code>\@ReturnAfterFi</code> .....	65, 70, <u>163</u>	<code>\@secondoftwo</code> .....
<code>\@car</code> .....	350	..... 161, 167, 253, 309, 339, 380
<code>\@cclv</code> .....	630, 672	<code>\@tempa</code> ..... 1156
<code>\@ehc</code> ..... 136, 143, 299, 513, 524, 736		<code>\@undefined</code> ..... 1156
<code>\@empty</code> .. 40, 50, 51, 111, 118, 125,		<code>\@</code> ..... 46, 62, 1019
185, 205, 213, 244, 285, 288,		
		<b>A</b>
		<code>\afterassignment</code> ..... 552

\AfterPackage .....	540	\HyColor@@IfRGB .....	315, 320
\annot ..	974, 979, 983, 988, 1012, 1017	\HyColor@@UseColor .....	187, 191
\AtBeginDocument .....	545	\HyColor@AttachfileColor .....	
\atfi@SETCMYKCOLORTest .....	968	.....	445, 974, 983, 1012
\atfi@SETGRAYCOLORTest .....	967	\HyColor@BookmarkColor ..	96, 785, 896
\atfi@SETRGBCOLORTest .....	966	\HyColor@CheckDot .....	41, 44
		\HyColor@CheckNum ..	324, 326, 328, 345
		\HyColor@DefSanitized ..	6, 154, 155, 160
		\HyColor@DetectPdfVersion .....	
		.....	424, 466, 478
		\HyColor@dot .....	344, 355
		\HyColor@ErrorModelNoXcolor .....	
		.....	112, 131, 229, 239, 400, 488
		\HyColor@ErrorSpecNoXcolor .....	
		.....	126, 138, 277, 418, 499
		\HyColor@FieldBColor .....	283
		\HyColor@FieldColor .....	284, 955
		\HyColor@HyperrefBorderColor ..	
		.....	384, 854, 866, 880
		\HyColor@HyperrefColor ..	200, 924, 929
		\HyColor@IfModel .....	
		.....	97, 145, 201, 211, 385, 480
		\HyColor@IfRGB .....	247, 304, 409
		\HyColor@IfXcolor ..	98, 120, 164,
			222, 232, 250, 386, 412, 481, 491
		\HyColor@MatchNum .....	364, 368, 369
		\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
		\HyColor@model@cmymk .....	
		.....	178, 179, 219, 260, 296, 477
		\HyColor@model@empty ..	172, 173, 212, 510
		\HyColor@model@Gray ..	180, 181, 221, 263
		\HyColor@model@gray .....	
		.....	106, 174, 175, 214, 224,
			226, 254, 265, 268, 292, 394, 465
		\HyColor@model@rgb .....	100,
			103, 122, 176, 177, 217, 234,
			236, 257, 271, 273, 294, 335,
			370, 388, 391, 414, 458, 483, 494
		\HyColor@NormalizeCommaCMYK .....	
		.....	86, 220, 261
		\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, 758
		\HyColor@one .....	343, 353
		\HyColor@PdfVersion .....	
			427, 429, 467, 479, 973, 982, 1011
		\HyColor@resultfalse .....	
		.....	332, 348, 357, 379, 451
		\HyColor@resulttrue .....	321, 376
		\HyColor@ReverseString ..	49, 54, 61, 66
		\HyColor@SpaceToComma .....	433, 455
		\HyColor@StripLeadingZeros ..	50, 71, 73

<code>\HyColor@temp</code> . . . . .	<code>\NoEmptyModel</code> . . . . .
. . . . . 79, 80, 81, 82, 83, 84, 87,	. . . . . 1027, 1051, 1052, 1089, 1090
88, 89, 90, 91, 92, 93, 94, 322,	<code>\nofiles</code> . . . . . 1099
323, 346, 347, 350, 351, 353, 355	
<code>\HyColor@TwoSpaces</code> . . . . . 319, 323	
<code>\HyColor@UseColor</code> . . . . . 182	
<code>\HyColor@values</code> . 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	
<code>\HyColor@WithModel</code> . . . . . 147, 153	
<code>\HyColor@WithoutModel</code> . . . . . 149, 158	
<code>\HyColor@XZeroOneThreeFour</code> . . . . .	
. . . . . 210, 283,	
286, 452, 817, 831, 844, 936, 949	
<code>\HyColor@zero</code> . . . . . 342, 351	
<b>I</b>	
<code>\ifdim</code> . . . . . 37	
<code>\ifError</code> . . . . . 994, 1002, 1021	
<code>\ifHyColor@result</code> . . . . .	
. . . . . 304, 325, 327, 334, 453	
<code>\ifin@</code> . . . . . 585, 599	
<code>\ifnum</code> . . . . . 45, 368, 467, 479	
<code>\ifx</code> . . . . . 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, 699, 833, 845, 868, 882,	
930, 950, 1001, 1019, 1133, 1138	
<code>\immediate</code> . . . . . 726	
<code>\in@</code> . . . . . 584, 598	
<code>\IncludeTests</code> . . . . . 717	
<code>\inline</code> . 976, 978, 985, 987, 1014, 1016	
<code>\inlinemacro</code> . . . . .	
. . . 974, 976, 983, 985, 1012, 1014	
<b>L</b>	
<code>\LogTests</code> . . . . . 718	
<b>M</b>	
<code>\makeatletter</code> . . . . . 720, 1115, 1154	
<code>\makeatother</code> . . . . . 1158	
<code>\MakeUppercase</code> . . . . . 1160	
<code>\Message</code> . . . . . 725, 726, 744, 745, 746	
<code>\MessageBreak</code> . . . . . 133, 134,	
140, 141, 794, 795, 805, 806,	
827, 828, 840, 841, 862, 863,	
876, 877, 1031, 1032, 1036, 1037	
<code>\ModelNoXcolor</code> . . . . . 1030, 1055	
<b>N</b>	
<code>\NeedsTeXFormat</code> . . . 2, 534, 1098, 1150	
<code>\newcommand</code> . . . . .	
722, 723, 725, 728, 1116, 1130, 1131	
<code>\newif</code> . . . . . 304, 994	
	<b>O</b>
	<code>\OptionName</code> . . . . . 896, 936, 949, 955
	<code>\OrgPackageError</code> . . . . . 998, 1006
	<b>P</b>
	<code>\PackageError</code> . . . . . 132,
	139, 299, 511, 522, 736, 998, 999
	<code>\PackageName</code> . . . . . 896, 936, 949, 955
	<code>\pdflastmatch</code> . . . . . 372, 373, 374
	<code>\pdfmatch</code> . . . . . 368, 710
	<code>\protect</code> . . . . . 696
	<code>\protected@edef</code> . . . . . 1155, 1157
	<code>\ProvidesFile</code> . . . . . 706, 709, 712
	<code>\ProvidesPackage</code> . . . . . 3, 535
	<b>R</b>
	<code>\rangeGray</code> . . . . . 641, 683
	<code>\rangeHSB</code> . . . . . 636, 678
	<code>\rangeRGB</code> . . . . . 625, 667
	<code>\remove@to@nnil</code> . . . 561, 573, 587, 601
	<code>\RequirePackage</code> . . . . . 5
	<code>\reserved@a</code> . . . . . 542,
	552, 557, 568, 582, 595, 610, 650
	<b>S</b>
	<code>\saved@errhelp</code> . . . . . 730, 738
	<code>\saved@errmessage</code> . . . . . 731, 743
	<code>\setbox</code> . . . . . 971, 1009
	<code>\space</code> . . . . . 319, 368, 369
	<code>\spec</code> . . . 974, 977, 983, 986, 1012, 1015
	<code>\SpecNoXColor</code> . . . . . 1035, 1058, 1059
	<code>\StartModel</code> . . . . . 1130, 1135, 1136
	<code>\StartValues</code> . . . . . 1131, 1135, 1136
	<code>\strip@prefix</code> . . . . . 372, 373, 374
	<b>T</b>
	<code>\Test</code> . . . . . 969,
	1041, 1042, 1044, 1046, 1048,
	1050, 1067, 1068, 1070, 1072,
	1074, 1076, 1077, 1078, 1079,
	1080, 1082, 1083, 1085, 1086, 1087
	<code>\test</code> 757, 761, 762, 763, 764, 765, 766,
	767, 768, 769, 770, 771, 772,
	773, 774, 775, 776, 777, 778,
	779, 780, 784, 788, 789, 790,
	791, 798, 809, 816, 820, 821,
	822, 823, 824, 853, 857, 858,
	859, 895, 899, 900, 901, 902,
	903, 904, 905, 906, 907, 908,
	909, 910, 911, 912, 913, 914,
	915, 916, 917, 918, 919, 923,
	927, 928, 935, 939, 940, 941,
	942, 943, 944, 945, 946, 947,
	948, 954, 958, 959, 960, 961, 962
	<code>\TestError</code> . . . . . 995, 1051, 1052,
	1053, 1056, 1058, 1059, 1089, 1090
	<code>\TestOptionName</code> . . . . . 723, 785,
	796, 807, 817, 832, 844, 854,
	867, 881, 975, 984, 1013, 1028, 1033



