The hycolor package

Heiko Oberdiek*
<heiko.oberdiek at googlemail.com>

2016/05/16 v1.8

Abstract

Package hycolor implements the color option stuff that is used by packages hyperref and bookmark. It is not intended as package for the user.

Contents

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

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 emtpy color setting without values. This applies to form field colors. The new A virtual color model `empty` is introduced for this purpose, e.g.

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

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

1.1 Summary

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

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

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

2 Implementation

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

2.1 Normalization

2.1.1 Sanitize value of color option

Procedure DefSanitized(cmd, value)

Param: *cmd* (macro)
Param: *value* (value of color option)

Result: *value* is expanded, sanitized, and stored in macro *cmd*.

Initialize active characters;
cmd := Expand *value*;
Sanitize *cmd*;

Sanitization means that the string does not contain any macros or special tokens. It consists of characters with catcode 12 (other). The only exception is the space with catcode 10 (space).

\HyColor@DefSanitized

```
6 \begingroup
7   \catcode`\!=13 %
8   \catcode`\:=13 %
9   \catcode`\-=13 %
10  \catcode`\+=13 %
11  \catcode`\;=13 %
12  \catcode`\"=13 %
13  \catcode`\>=13 %
14  \edef\x{%
15    \def\noexpand!{\string!}%
16    \def\noexpand:{\string:}%
17    \def\noexpand-{\string-}%
```

```
    \def\noexpand+{\string+}%
    \def\noexpand;{\string;}%
    \def\noexpand"{\string"}%
    \def\noexpand>{\string>}%
  }%
  \def\y#1{\endgroup
    \def\HyColor@DefSanitized##1##2{%
      \begingroup
        \csname @safe@activestrue\endcsname
        #1%
        \edef\x{\endgroup
          \def\noexpand##1{##2}%
        }%
      \x
      \@onelevel@sanitize##1%
    }%
  }%
\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`

```
\def\HyColor@NormalizeNum#1#2{%
  \ifdim#1pt<\z@
    \def#2{0}%
  \else
    \edef#2{\zap@space#1 \@empty}%
    \expandafter\HyColor@CheckDot#2..\@nil#2%
  \fi
}
\def\HyColor@CheckDot#1.#2.#3\@nil#4{%
  \ifnum0#1<\@ne
    \ifx\\#2\\%
      \def#4{0}%
    \else
      \edef#4{\HyColor@ReverseString#2\@nil{}}%
```

```
      \edef#4{\expandafter\HyColor@StripLeadingZeros#4\@empty}%
      \ifx#4\@empty
        \def#4{0}%
      \else
        \edef#4{.\expandafter\HyColor@ReverseString#4\@nil{}}%
      \fi
    \fi
  \else
    \def#4{1}%
  \fi
}
\def\HyColor@ReverseString#1#2\@nil#3{%
  \ifx\\#2\\%
    #1#3%
  \else
    \@ReturnAfterFi{%
      \HyColor@ReverseString#2\@nil{#1#3}%
    }%
  \fi
}
\long\def\@ReturnAfterFi#1\fi{\fi#1}
\def\HyColor@StripLeadingZeros#1{%
  \ifx#10%
    \expandafter\HyColor@StripLeadingZeros
  \else
    #1%
  \fi
}
```

\HyColor@NormalizeCommaRGB

```
\def\HyColor@NormalizeCommaRGB#1,#2,#3\@nil#4{%
  \HyColor@NormalizeNum{#1}\HyColor@temp
  \let#4\HyColor@temp
  \HyColor@NormalizeNum{#2}\HyColor@temp
  \edef#4{#4 \HyColor@temp}%
  \HyColor@NormalizeNum{#3}\HyColor@temp
  \edef#4{#4 \HyColor@temp}%
}
```

\HyColor@NormalizeCommaCMYK

```
\def\HyColor@NormalizeCommaCMYK#1,#2,#3,#4\@nil#5{%
  \HyColor@NormalizeNum{#1}\HyColor@temp
  \let#5\HyColor@temp
  \HyColor@NormalizeNum{#2}\HyColor@temp
  \edef#5{#5 \HyColor@temp}%
  \HyColor@NormalizeNum{#3}\HyColor@temp
  \edef#5{#5 \HyColor@temp}%
  \HyColor@NormalizeNum{#4}\HyColor@temp
  \edef#5{#5 \HyColor@temp}%
}
```

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:

`rgbcolor=`⟨*r*⟩ ⟨*g*⟩ ⟨*b*⟩ ≡ `color=[rgb]{`⟨*r*⟩`,`⟨*g*⟩`,`⟨*b*⟩`}`

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

Procedure BookmarkColor(value, cmd, package, option)

Param: *value* (value of option color)
Param: *cmd* (macro for result)
Param: *package* (package name for error message)
Param: *option* (option name for error message)

```
switch value do
    case empty do
        cmd ← no color;
    end
    case with model do
        if with xcolor then
            cmd ← ConvertToRGB(model, values);
        else
            if model = rgb then
                cmd ← values as normalized values;
            else if model = gray then
                cmd ← values as normalized tripled values;
            else
                error;
            end
        end
    end
    otherwise do
        if with xcolor then
            (model, values ← get model and values;
            cmd ← ConvertToRGB(model, values);
        else
            error;
        end
    end
end
```

```
\def\HyColor@BookmarkColor#1#2#3#4{%
  \HyColor@IfModel{#1}{%
    \HyColor@IfXcolor{%
      \convertcolorspec\HyColor@model\HyColor@values
                       \HyColor@model@rgb#2%
      \expandafter\HyColor@NormalizeCommaRGB#2\@nil#2%
    }{%
      \ifx\HyColor@model\HyColor@model@rgb
        \expandafter\HyColor@NormalizeCommaRGB\HyColor@values\@nil#2%
      \else
        \ifx\HyColor@model\HyColor@model@gray
          \expandafter\HyColor@NormalizeNum
              \expandafter{\HyColor@values}#2%
          \edef#2{#2 #2 #2}%
        \else
          \let#2\@empty
          \HyColor@ErrorModelNoXcolor{#3}{#4}%
        \fi
      \fi
    }%
  }{%
    \let#2\HyColor@values
    \ifx#2\@empty
    \else
      \HyColor@IfXcolor{%
        \extractcolorspec{#1}#2%
        \expandafter\convertcolorspec#2\HyColor@model@rgb#2%
        \expandafter\HyColor@NormalizeCommaRGB#2\@nil#2%
      }{%
        \let#2\@empty
        \HyColor@ErrorSpecNoXcolor{#3}{#4}%
      }%
    \fi
  }%
}
\def\HyColor@ErrorModelNoXcolor#1#2{%
  \PackageError{#1}{%
    Color model `\HyColor@model' is not supported\MessageBreak
    without package `xcolor' in\MessageBreak
    `#2=[\HyColor@model]{\HyColor@values}'%
  }\@ehc
}
\def\HyColor@ErrorSpecNoXcolor#1#2{%
  \PackageError{#1}{%
    This color specification is not supported\MessageBreak
    without package `xcolor' in\MessageBreak
    `#2=\HyColor@values'%
  }\@ehc
}
\def\HyColor@IfModel#1{%
  \@ifnextchar[{%
    \HyColor@WithModel
  }{%
    \HyColor@WithoutModel
  }%
  #1\@nil
}
```

```
\def\HyColor@WithModel[#1]#2\@nil{%
  \HyColor@DefSanitized\HyColor@model{#1}%
  \HyColor@DefSanitized\HyColor@values{#2}%
  \@firstoftwo
}
\def\HyColor@WithoutModel#1\@nil{%
  \let\HyColor@model\relax
  \HyColor@DefSanitized\HyColor@values{#1}%
  \@secondoftwo
}
```

2.4 Utils

\@ReturnAfterFi

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

\HyColor@IfXcolor

```
\def\HyColor@IfXcolor{%
  \begingroup\expandafter\expandafter\expandafter\endgroup
  \expandafter\ifx\csname convertcolorspec\endcsname\relax
    \expandafter\@secondoftwo
  \else
    \expandafter\@firstoftwo
  \fi
}
```

```
\def\HyColor@model@empty{empty}
\@onelevel@sanitize\HyColor@model@empty
\def\HyColor@model@gray{gray}
\@onelevel@sanitize\HyColor@model@gray
\def\HyColor@model@rgb{rgb}
\@onelevel@sanitize\HyColor@model@rgb
\def\HyColor@model@cmyk{cmyk}
\@onelevel@sanitize\HyColor@model@cmyk
\def\HyColor@model@Gray{Gray}
\@onelevel@sanitize\HyColor@model@Gray
```

2.5 Package hyperref

2.5.1 Options Hyp.*color

```
\def\HyColor@UseColor#1{%
  \ifx#1\relax
  \else
    \ifx#1\@empty
    \else
      \expandafter\HyColor@@UseColor#1\@nil
    \fi
  \fi
}
\def\HyColor@@UseColor{%
  \@ifnextchar[\HyColor@@@UseColor\HyColor@@@@UseColor
}
\def\HyColor@@@UseColor[#1]#2\@nil{%
  \color[{#1}]{#2}%
}
\def\HyColor@@@@UseColor#1\@nil{%
  \color{#1}%
```

```
}
```

Procedure HyperrefColor(value, cmd)

Param: *value* (value of the option)
Param: *cmd* (macro for result)

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

```
\def\HyColor@HyperrefColor#1#2{%
  \HyColor@IfModel{#1}{%
    \edef#2{[{\HyColor@model}]{\HyColor@values}}%
  }{%
    \let#2\HyColor@values
    \ifx#2\@empty
      \let#2\relax
    \fi
  }%
}
```

2.5.2 Generic algorithm

Procedure Algorithm X0134(value, cmd, package, option)

Param: *value* (value of the option)
Param: *cmd* (macro for result)
Param: *package* (package name for error message)
Param: *option* (option name for error message)

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

\HyColor@XZeroOneThreeFour

```
\def\HyColor@XZeroOneThreeFour#1#2#3#4{%
  \HyColor@IfModel{#1}{%
    \ifx\HyColor@model\HyColor@model@empty
      \let#2\@empty
    \else\ifx\HyColor@model\HyColor@model@gray
      \expandafter\HyColor@NormalizeNum
          \expandafter{\HyColor@values}#2%
    \else\ifx\HyColor@model\HyColor@model@rgb
      \expandafter\HyColor@NormalizeCommaRGB\HyColor@values\@nil#2%
    \else\ifx\HyColor@model\HyColor@model@cmyk
      \expandafter\HyColor@NormalizeCommaCMYK\HyColor@values\@nil#2%
    \else\ifx\HyColor@model\HyColor@model@Gray
      \HyColor@IfXcolor{%
        \convertcolorspec\HyColor@model\HyColor@values
                         \HyColor@model@gray#2%
        \expandafter\HyColor@NormalizeNum\expandafter{#2}#2%
        \let\HyColor@model\HyColor@model@gray
      }{%
        \let#2\relax
        \HyColor@ErrorModelNoXcolor{#3}{#4}%
      }%
    \else
      \HyColor@IfXcolor{%
        \convertcolorspec\HyColor@model\HyColor@values
                         \HyColor@model@rgb#2%
        \expandafter\HyColor@NormalizeCommaRGB#2\@nil#2%
        \let\HyColor@model\HyColor@model@rgb
      }{%
        \let#2\relax
        \HyColor@ErrorModelNoXcolor{#3}{#4}%
      }%
    \fi\fi\fi\fi\fi
  }{%
    \let#2\HyColor@values
    \ifx#2\@empty
      \let#2\relax
    \else
      \expandafter\HyColor@IfRGB\expandafter{\HyColor@values}{%
        \expandafter\HyColor@NormalizeCommaRGB\HyColor@values\@nil#2%
      }{%
        \HyColor@IfXcolor{%
          \expandafter\extractcolorspec\expandafter{\HyColor@values}#2%
          \edef\HyColor@model{\expandafter\@firstoftwo#2}%
          \edef\HyColor@values{\expandafter\@secondoftwo#2}%
          \ifx\HyColor@model\HyColor@model@gray
            \expandafter\HyColor@NormalizeNum\expandafter
                {\HyColor@values}#2%
          \else\ifx\HyColor@model\HyColor@model@rgb
            \expandafter\HyColor@NormalizeCommaRGB
                \HyColor@values\@nil#2%
          \else\ifx\HyColor@model\HyColor@model@cmyk
            \expandafter\HyColor@NormalizeCommaCMYK
                \HyColor@values\@nil#2%
          \else\ifx\HyColor@model\HyColor@model@Gray
            \convertcolorspec\HyColor@model\HyColor@values
                \HyColor@model@gray#2%
            \expandafter\HyColor@NormalizeNum\expandafter
```

```
                {\HyColor@values}#2%
            \let\HyColor@model\HyColor@model@gray
          \else
            \convertcolorspec\HyColor@model\HyColor@values
                \HyColor@model@rgb#2%
            \expandafter\HyColor@NormalizeCommaRGB#2\@nil#2%
            \let\HyColor@model\HyColor@model@rgb
          \fi\fi\fi\fi
        }{%
          \let#2\relax
          \HyColor@ErrorSpecNoXcolor{#3}{#4}%
        }%
      }%
    \fi
  }%
}
```

2.5.3 Field options

\HyColor@FieldBColor

```
\let\HyColor@FieldBColor\HyColor@XZeroOneThreeFour
```

\HyColor@FieldColor

```
\def\HyColor@FieldColor#1#2#3#4{%
  \let\HyColor@model\@empty
  \HyColor@XZeroOneThreeFour{#1}{#2}{#3}{#4}%
  \ifx#2\relax
    \let#2\@empty
  \else
    \ifx#2\@empty
    \else
      \ifx\HyColor@model\HyColor@model@gray
        \edef#2{#2 g}%
      \else\ifx\HyColor@model\HyColor@model@rgb
        \edef#2{#2 rg}%
      \else\ifx\HyColor@model\HyColor@model@cmyk
        \edef#2{#2 k}%
      \else
        \PackageError{#3}{Internal error: unsupported color model}\@ehc
      \fi\fi\fi
    \fi
  \fi
}
```

2.5.4 Detection for naked RGB values

\HyColor@IfRGB

```
\newif\ifHyColor@result
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname pdfmatch\endcsname\relax
  \expandafter\@firstoftwo
\else
  \expandafter\@secondoftwo
\fi
{%
  \begingroup
    \def\x#1{\endgroup
      \def\HyColor@IfRGB##1{%
```

```
      \HyColor@@IfRGB##1#1#1#1\@nil
    }%
  }%
\x{ }%
\edef\HyColor@TwoSpaces{\space\space}%
\def\HyColor@@IfRGB#1 #2 #3 #4\@nil{%
  \HyColor@resulttrue
  \def\HyColor@temp{#4}%
  \ifx\HyColor@temp\HyColor@TwoSpaces
    \HyColor@CheckNum{#1}%
    \ifHyColor@result
      \HyColor@CheckNum{#2}%
      \ifHyColor@result
        \HyColor@CheckNum{#3}%
      \fi
    \fi
  \else
    \HyColor@resultfalse
  \fi
  \ifHyColor@result
    \let\HyColor@model\HyColor@model@rgb
    \edef\HyColor@values{#1,#2,#3}%
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi
}%
\def\HyColor@zero{0}%
\def\HyColor@one{1}%
\def\HyColor@dot{.}%
\def\HyColor@CheckNum#1{%
  \def\HyColor@temp{#1}%
  \ifx\HyColor@temp\@empty
    \HyColor@resultfalse
  \else
    \edef\HyColor@temp{\@car#1\@nil}%
    \ifx\HyColor@temp\HyColor@zero
    \else
      \ifx\HyColor@temp\HyColor@one
      \else
        \ifx\HyColor@temp\HyColor@dot
        \else
          \HyColor@resultfalse
        \fi
      \fi
    \fi
  \fi
}%
}{%
  \def\HyColor@MatchNum{%
    (0*1\string\.0*|0*1|0+\string\.?[0-9]*|\string\.[0-9]+)%
  }%
  \def\HyColor@IfRGB#1{%
    \ifnum\pdfmatch{^\HyColor@MatchNum\space\HyColor@MatchNum
        \space\HyColor@MatchNum$}{#1}>\z@
      \let\HyColor@model\HyColor@model@rgb
      \edef\HyColor@values{%
        \expandafter\strip@prefix\pdflastmatch1,%
```

```
        \expandafter\strip@prefix\pdflastmatch2,%
        \expandafter\strip@prefix\pdflastmatch3%
      }%
      \HyColor@resulttrue
      \expandafter\@firstoftwo
    \else
      \HyColor@resultfalse
      \expandafter\@secondoftwo
    \fi
  }%
}
```

2.5.5 Options *bordercolor

Procedure HyperrefBorderColor(value, cmd, package, option)

Param: *value* (value of the option)
Param: *cmd* (macro for result)
Param: *package, option* (package and option for error message)

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

\HyColor@HyperrefBorderColor

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

```
\HyColor@IfModel{#1}{%
  \HyColor@IfXcolor{%
    \convertcolorspec\HyColor@model\HyColor@values
                     \HyColor@model@rgb#2%
    \expandafter\HyColor@NormalizeCommaRGB#2\@nil#2%
  }{%
    \ifx\HyColor@model\HyColor@model@rgb
      \expandafter\HyColor@NormalizeCommaRGB\HyColor@values\@nil#2%
    \else
      \ifx\HyColor@model\HyColor@model@gray
        \expandafter\HyColor@NormalizeNum
            \expandafter{\HyColor@values}#2%
        \edef#2{#2 #2 #2}%
      \else
        \let#2\relax
        \HyColor@ErrorModelNoXcolor{#3}{#4}%
      \fi
    \fi
  }%
}{%
  \let#2\HyColor@values
  \ifx#2\@empty
    \let#2\relax
  \else
    \expandafter\HyColor@IfRGB\expandafter{\HyColor@values}{%
      \expandafter\HyColor@NormalizeCommaRGB\HyColor@values\@nil#2%
    }{%
      \HyColor@IfXcolor{%
        \extractcolorspec{#1}#2%
        \expandafter\convertcolorspec#2\HyColor@model@rgb#2%
        \expandafter\HyColor@NormalizeCommaRGB#2\@nil#2%
      }{%
        \let#2\relax
        \HyColor@ErrorSpecNoXcolor{#3}{#4}%
      }%
    }%
  \fi
}%
}
```

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`

```
\def\HyColor@DetectPdfVersion{%
```

```
  \begingroup\expandafter\expandafter\expandafter\endgroup
  \expandafter\ifx\csname Hy@pdfversion\endcsname\relax
    \global\chardef\HyColor@PdfVersion=0 %
  \else
    \global\chardef\HyColor@PdfVersion=\Hy@pdfversion\relax
  \fi
  \global\let\HyColor@DetectPdfVersion\relax
}
```

\HyColor@SpaceToComma

```
\def\HyColor@SpaceToComma#1 #2\@nil{%
  #1%
  \ifx\relax#2\relax
    \expandafter\@gobble
  \else
    ,%
    \expandafter\@firstofone
  \fi
  {%
    \HyColor@SpaceToComma#2\@nil
  }%
}%
```

\HyColor@AttachfileColor

```
\def\HyColor@AttachfileColor#1#2#3#4#5#6{%
  \def#2{#1}%
  \ifx#2\@empty
    \let#3\@gobble
    \let#4\@empty
  \else
    \HyColor@resultfalse
    \HyColor@XZeroOneThreeFour{#1}#3{#5}{#6}%
    \ifHyColor@result
      \edef#2{%
        [rgb]{\expandafter\HyColor@SpaceToComma#3 \@nil}%
      }%
    \fi
    \ifx\HyColor@model\HyColor@model@rgb
      \edef#4{/C[#3]}% hash-ok
      \edef#3##1{%
        #3 %
        \noexpand\csname atfi@SETRGBCOLOR##1\noexpand\endcsname
      }%
    \else
      \ifx\HyColor@model\HyColor@model@gray
        \HyColor@DetectPdfVersion
        \ifnum\HyColor@PdfVersion<7 %
          \edef#4{/C[#3 #3 #3]}% hash-ok
        \else
          \edef#4{/C[#3]}% hash-ok
        \fi
        \edef#3##1{%
          #3 %
          \noexpand\csname atfi@SETGRAYCOLOR##1\noexpand\endcsname
        }%
      \else
        \ifx\HyColor@model\HyColor@model@cmyk
          \HyColor@DetectPdfVersion
```

```
          \ifnum\HyColor@PdfVersion<7 %
            \HyColor@IfModel{#1}{%
              \HyColor@IfXcolor{%
                \convertcolorspec\HyColor@model\HyColor@values
                                 \HyColor@model@rgb#4%
                \expandafter\HyColor@NormalizeCommaRGB#4\@nil#4%
                \edef#4{/C[#4]}% hash-ok
              }{%
                \let#4\@empty
                \HyColor@ErrorModelNoXcolor{#5}{#6}%
              }%
            }{%
              \HyColor@IfXcolor{%
                \extractcolorspec{#1}#4%
                \expandafter\convertcolorspec#4%
                    \HyColor@model@rgb#4%
                \expandafter\HyColor@NormalizeCommaRGB#4\@nil#4%
                \edef#4{/C[#4]}% hash-ok
              }{%
                \let#4\@empty
                \HyColor@ErrorSpecNoXcolor{#5}{#6}%
              }%
            }%
          \else
            \edef#4{/C[#3]}% hash-ok
          \fi
          \edef#3##1{%
            #3 %
            \noexpand\csname atfi@SETCMYKCOLOR##1\noexpand\endcsname
          }%
        \else
          \ifx\HyColor@model\HyColor@model@empty
            \PackageError{#5}{%
              Color model 'empty' is not permitted for option '#6'%
            }\@ehc
            \let#2\@empty
            \let#3\@gobble
            \let#4\@empty
          \else
            \ifx\HyColor@model\relax % (missing xcolor)
              \let#3\@gobble
              \let#4\@empty
            \else
              \PackageError{#5}{%
                Internal error: unsupported color model%
              }\@ehc
            \fi
          \fi
        \fi
      \fi
    \fi
  \fi
}
```

⟨/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 found 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 scrlfile is loaded. As last resort `\AtBeginDocument` is used.

```
⟨*xcolor⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{xcolor-patch}[2016/05/16 xcolor patch]
\RequirePackage{hopatch}[2016/05/16]
\hopatch@AfterPackage{xcolor}{%
```

`\XC@ifxcase`

```
  \long\def\reserved@a#1#2#3{%
    \long\def\@@tmp##1##2{%
      \ifx#1##1%
        \toks@{##2}%
        \expandafter\remove@to@nnil
      \else
        \expandafter\@@tmp
      \fi
    }%
    \@@tmp#2#1{#3}\@nnil\the\toks@
  }%
  \ifx\XC@ifxcase\reserved@a
    \long\def\XC@ifxcase#1#2#3{%
      \long\def\XC@if@##1##2{%
        \ifx#1##1%
          \toks@{##2}%
          \expandafter\remove@to@nnil
        \else
          \expandafter\XC@if@
        \fi
      }%
      \XC@if@#2#1{#3}\@nnil
      \the\toks@
    }%
  \fi
```

`\XC@ifcase`

```
  \long\def\reserved@a#1#2#3{%
    \long\def\@@tmp##1##2{%
      \@expandtwoargs\in@{,#1,}{,##1,}%
      \ifin@
        \toks@{##2}%
        \expandafter\remove@to@nnil
      \else
        \expandafter\@@tmp
      \fi
    }%
    \@@tmp#2{#1}{#3}\@nnil
    \the\toks@
  }%
```

```
\ifx\XC@ifcase\reserved@a
  \long\def\XC@ifcase#1#2#3{%
    \long\def\XC@if@##1##2{%
      \@expandtwoargs\in@{,#1,}{,##1,}%
      \ifin@
        \toks@{##2}%
        \expandafter\remove@to@nnil
      \else
        \expandafter\XC@if@
      \fi
    }%
    \XC@if@#2{#1}{#3}\@nnil
    \the\toks@
  }%
\fi
```

\XC@cnv@gray

```
\def\reserved@a#1,{%
  \XC@ifxcase\tm{%
    \XC@mod@rgb{%
      \XC@calcN{#1}\@@tmp
      \edef\@@tmp{\@@tmp,\@@tmp,\@@tmp}%
    }%
    \XC@mod@cmy{%
      \XC@calcC{#1}\@@tmp
      \edef\@@tmp{\@@tmp,\@@tmp,\@@tmp}%
    }%
    \XC@mod@cmyk{%
      \XC@calcC{#1}\@@tmp
      \edef\@@tmp{0,0,0,\@@tmp}%
    }%
    \XC@mod@RGB{%
      \edef\@@scl{\rangeRGB}%
      \XC@calcM{#1}\@@tmp
      \edef\@@tmp{\@@tmp,\@@tmp,\@@tmp}%
    }%
    \XC@mod@HTML{%
      \edef\@@scl{\@cclv}%
      \XC@calcM{#1}\@@tmp
      \XC@calcH\@@tmp\@@tmp
      \edef\@@tmp{\@@tmp\@@tmp\@@tmp}%
    }%
    \XC@mod@HSB{%
      \edef\@@scl{\rangeHSB}%
      \XC@calcM{#1}\@@tmp
      \edef\@@tmp{0,0,\@@tmp}%
    }%
    \XC@mod@Gray{%
      \edef\@@scl{\rangeGray}%
      \XC@calcM{#1}\@@tmp
    }%
  }%
  {%
    \XC@calcN{#1}\@@tmp
    \edef\@@tmp{0,0,\@@tmp}%
  }%
}%
\ifx\XC@cnv@gray\reserved@a
```

```
  \def\XC@cnv@gray#1,{%
    \XC@ifxcase\tm{%
      \XC@mod@rgb{%
        \XC@calcN{#1}\@@tmp
        \edef\@@tmp{\@@tmp,\@@tmp,\@@tmp}%
      }%
      \XC@mod@gray{}%
      \XC@mod@cmy{%
        \XC@calcC{#1}\@@tmp
        \edef\@@tmp{\@@tmp,\@@tmp,\@@tmp}%
      }%
      \XC@mod@cmyk{%
        \XC@calcC{#1}\@@tmp
        \edef\@@tmp{0,0,0,\@@tmp}%
      }%
      \XC@mod@RGB{%
        \edef\@@scl{\rangeRGB}%
        \XC@calcM{#1}\@@tmp
        \edef\@@tmp{\@@tmp,\@@tmp,\@@tmp}%
      }%
      \XC@mod@HTML{%
        \edef\@@scl{\@cclv}%
        \XC@calcM{#1}\@@tmp
        \XC@calcH\@@tmp\@@tmp
        \edef\@@tmp{\@@tmp\@@tmp\@@tmp}%
      }%
      \XC@mod@HSB{%
        \edef\@@scl{\rangeHSB}%
        \XC@calcM{#1}\@@tmp
        \edef\@@tmp{0,0,\@@tmp}%
      }%
      \XC@mod@Gray{%
        \edef\@@scl{\rangeGray}%
        \XC@calcM{#1}\@@tmp
      }%
    }%
    {%
      \XC@calcN{#1}\@@tmp
      \edef\@@tmp{0,0,\@@tmp}%
    }%
  }%
\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
```

```
\@ifundefined{XC@frameb@x }{%
  \expandafter\let\csname XC@frameb@x \endcsname\XC@frameb@x
  \edef\XC@frameb@x{%
    \noexpand\protect
    \expandafter\noexpand\csname XC@frameb@x \endcsname
```

```
    }%
    \expandafter\ifx\csname XC@frameb@x \endcsname\@frameb@x
      \let\@frameb@x\XC@frameb@x
    \fi
  }{}%
}
```

⟨/xcolor⟩

3 Test

⟨*test1⟩

```
\ProvidesFile{hycolor-test1.tex}[2016/05/16 test file 1]
```

⟨/test1⟩

⟨*test2⟩

```
\ProvidesFile{hycolor-test2.tex}[2016/05/16 test file 2]
\let\pdfmatch\relax
```

⟨/test2⟩

⟨test3⟩`\ProvidesFile{hycolor-test3.tex}[2016/05/16 test file 3]`

⟨*test⟩

```
\documentclass{article}

\usepackage{qstest}
\IncludeTests{*}
\LogTests{log}{*}{*}

\makeatletter

\newcommand*{\TestPackageName}{test-package}
\newcommand*{\TestOptionName}{test-option}

\newcommand\Message{}
\def\Message#1#{\immediate\write16}

\newcommand*{\ExpectError}[2]{%
  \begingroup
    \global\let\saved@errhelp\errhelp
    \global\let\saved@errmessage\errmessage
    \let\errhelp\@gobble
    \def\errmessage##1{%
      \xdef\@ExpectErrorMessage{##1}%
    }%
    \PackageError\TestPackageName{#1}\@ehc
    \def\errhelp##1{%
      \global\let\errhelp\saved@errhelp
    }%
    \global\let\@ResultErrorMessage\@empty
    \def\errmessage##1{%
      \xdef\@ResultErrorMessage{##1}%
      \global\let\errmessage\saved@errmessage
      % \Message{[ ##1}%
      % \Message{] (ignored error)}%
      % \Message{}%
    }%
    #2%
  \endgroup
```

```
  \Expect*{\@ResultErrorMessage}*{\@ExpectErrorMessage}%
}
\usepackage{scrlfile}
\usepackage{hycolor}[2016/05/16]
⟨/test⟩
⟨*test1⟩
\begin{qstest}{NumNormalize}{num, normalize}
  \def\test#1#2{%
    \HyColor@NormalizeNum{#1}\cmd
    \Expect*{\cmd}{#2}%
  }%
  \test{0}{0}%
  \test{000}{0}%
  \test{-1}{0}%
  \test{ 0 }{0}%
  \test{1.1}{1}%
  \test{100}{1}%
  \test{00100}{1}%
  \test{99.99}{1}%
  \test{0.0}{0}%
  \test{00.00}{0}%
  \test{0.}{0}%
  \test{.0}{0}%
  \test{0.1}{.1}%
  \test{0.10}{.1}%
  \test{0.1000}{.1}%
  \test{.1000}{.1}%
  \test{0.01}{.01}%
  \test{0.01010}{.0101}%
  \test{.0000000001}{.0000000001}%
  \test{.9999999999}{.9999999999}%
\end{qstest}

\begin{qstest}{BookmarkColor without xcolor}{bookmark, noxcolor}
  \def\test#1#2{%
    \HyColor@BookmarkColor{#1}\cmd\TestPackageName\TestOptionName
    \Expect*{\cmd}{#2}%
  }%
  \test{[rgb]{1,0,0}}{1 0 0}%
  \test{[gray]{0.10}}{.1 .1 .1}%
  \test{}{}%
  \test{[rgb]{ 1 , 1 , 0 }}{1 1 0}%
  \def\errortest[#1]#2{%
    \ExpectError{%
      Color model ‘#1’ is not supported\MessageBreak
      without package ‘xcolor’ in\MessageBreak
      ‘\TestOptionName=[#1]{#2}’% hash-ok
    }{%
      \test{[#1]{#2}}{}% hash-ok
    }%
  }%
  \errortest[cmyk]{1,0,0,0}%
  \errortest[empty]{}%
  \def\errortest#1{%
    \ExpectError{%
      This color specification is not supported\MessageBreak
      without package ‘xcolor’ in\MessageBreak
      ‘\TestOptionName=#1’%
```

```
    }{%
      \test{#1}{}%
    }%
  }%
\end{qstest}
⟨/test1⟩
⟨*test1 | test2⟩
\begin{qstest}{X0134 without xcolor}{X0134, noxcolor}
  \def\test#1#2{%
    \HyColor@XZeroOneThreeFour{#1}\cmd\TestPackageName\TestOptionName
    \Expect*{\cmd}{#2}%
  }%
  \test{[empty]{}}{}%
  \test{[rgb]{1,0,0}}{1 0 0}%
  \test{[gray]{0.10}}{.1}%
  \test{[cmyk]{0,1,0,0}}{0 1 0 0}%
  \test{[rgb]{ 1 , 1 , 0 }}{1 1 0}%
  \def\errortest[#1]#2{%
    \ExpectError{%
      Color model `#1' is not supported\MessageBreak
      without package `xcolor' in\MessageBreak
      `test-option=[#1]{#2}'% hash-ok
    }{%
      \HyColor@XZeroOneThreeFour{[{#1}]{#2}}\cmd
          \TestPackageName\TestOptionName
      \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
    }%
  }%
  \errortest[Gray]{10}%
  \errortest[cmy]{1,0,0}%
  \def\errortest#1{%
    \ExpectError{%
      This color specification is not supported\MessageBreak
      without package `xcolor' in\MessageBreak
      `test-option=#1'%
    }{%
      \HyColor@XZeroOneThreeFour{#1}\cmd\TestPackageName\TestOptionName
      \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
    }%
  }%
  \errortest{yellow}%
\end{qstest}

\begin{qstest}{HyperrefBorderColor without xcolor}%
              {hyperef bordercolor, noxcolor}%
  \def\test#1#2{%
    \HyColor@HyperrefBorderColor{#1}\cmd\TestPackageName\TestOptionName
    \Expect*{\cmd}{#2}%
  }%
  \test{[rgb]{1,0,0}}{1 0 0}%
  \test{[gray]{0.10}}{.1 .1 .1}%
  \test{[rgb]{ 1 , 1 , 0 }}{1 1 0}%
  \def\errortest[#1]#2{%
    \ExpectError{%
      Color model `#1' is not supported\MessageBreak
      without package `xcolor' in\MessageBreak
      `test-option=[#1]{#2}'% hash-ok
    }{%
```

```
      \HyColor@HyperrefBorderColor{[{#1}]{#2}}\cmd
          \TestPackageName\TestOptionName
      \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
    }%
  }%
  \errortest[Gray]{10}%
  \errortest[cmy]{1,0,0}%
  \errortest[cmyk]{0,1,0,0}%
  \def\errortest#1{%
    \ExpectError{%
      This color specification is not supported\MessageBreak
      without package 'xcolor' in\MessageBreak
      'test-option=#1'%
    }{%
      \HyColor@HyperrefBorderColor{#1}\cmd
          \TestPackageName\TestOptionName
      \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
    }%
  }%
  \errortest{yellow}%
\end{qstest}
⟨/test1 | test2⟩
⟨*test1 | test2⟩
\usepackage{xcolor}
\definecolor[named]{MyGreen}{rgb}{0,0.7,0}
\definecolor{mygreen}{named}{MyGreen}
⟨/test1 | test2⟩
⟨*test1⟩
\begin{qstest}{BookmarkColor with xcolor}{bookmark, xcolor}
  \def\test#1#2{%
    \HyColor@BookmarkColor{#1}\cmd\PackageName\OptionName
    \Expect*{\cmd}{#2}%
  }%
  \test{[rgb]{1,0,0}}{1 0 0}%
  \test{[gray]{0.10}}{.1 .1 .1}%
  \test{}{}%
  \test{[rgb]{ 1 , 1 , 0 }}{1 1 0}%
  \test{[cmyk]{1,0,0,0}}{0 1 1}%
  \test{red}{1 0 0}%
  \test{cyan}{0 1 1}%
  \test{red!40!blue}{.4 0 .6}%
  \test{[Gray]{10}}{.66667 .66667 .66667}%
  \test{[RGB]{100,200,50}}{.39217 .78432 .19609}%
  \test{[wave]{363}}{.00316 0 .00316}%
  \test{[wave]814}{.00797 0 0}%
  \test{[HSB]{100,200,50}}{.03473 .20833 .12152}%
  \test{[HTML]{A800FF}}{.65881 0 1}%
  \test{[cmy]{.3,.5,.2}}{.7 .5 .8}%
  \test{[cmyk]{.3,.5,.2,.1}}{.6 .4 .7}%
  \test{[hsb]{.3,.5,.2}}{.12 .2 .1}%
  \test{[Hsb]{120,.5,.2}}{.1 .2 .1}%
  \test{[tHsb]{120,.5,.2}}{.2 .2 .1}%
  \test{[named]{MyGreen}}{0 .7 0}%
  \test{mygreen}{0 .7 0}%
\end{qstest}

\begin{qstest}{HyperrefColor}{hyperref, color}
  \def\test#1#2{%
```

```
    \HyColor@HyperrefColor{#1}\cmd
    \Expect*{\cmd}{#2}%
  }%
  \test{red}{red}%
  \test{[rgb]{1,0,0}}{[{rgb}]{1,0,0}}%
  \HyColor@HyperrefColor{}\cmd
  \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
\end{qstest}
⟨/test1⟩
⟨*test1 | test2⟩
\begin{qstest}{X0134 with xcolor}{hyperref, X0134, xcolor}
  \def\test#1#2{%
    \HyColor@XZeroOneThreeFour{#1}\cmd\PackageName\OptionName
    \Expect*{\cmd}{#2}%
  }%
  \test{[empty]{}}{}%
  \test{[gray]{0.1}}{.1}%
  \test{[rgb]{1,0.5,0.0}}{1 .5 0}%
  \test{[cmyk]{0,1,0,0.5}}{0 1 0 .5}%
  \test{[Gray]{10}}{.66667}%
  \test{red}{1 0 0}%
  \test{1 0 0}{1 0 0}%
  \test{001.0 .23 0}{1 .23 0}%
  \test{[named]{MyGreen}}{0 .7 0}%
  \test{mygreen}{0 .7 0}%
  \HyColor@XZeroOneThreeFour{}\cmd\PackageName\OptionName
  \Expect{true}*{\ifx\cmd\relax true\else false\fi}%
\end{qstest}

\begin{qstest}{FieldColor}{hyperref, field, FieldColor}
  \def\test#1#2{%
    \HyColor@FieldColor{#1}\cmd\PackageName\OptionName
    \Expect*{\cmd}{#2}%
  }%
  \test{}{}%
  \test{[gray]{0.7}}{.7 g}%
  \test{[rgb]{1,0,0}}{1 0 0 rg}%
  \test{[cmyk]{0,1,0,0}}{0 1 0 0 k}%
  \test{[cmy]{.5,.4,.3}}{.5 .6 .7 rg}%
\end{qstest}
⟨/test1 | test2⟩
```

3.1 Test for package **attachfile2**

```
⟨*test3⟩
\def\atfi@SETRGBCOLORtest{set-rgb}
\def\atfi@SETGRAYCOLORtest{set-gray}
\def\atfi@SETCMYKCOLORtest{set-cmyk}
\def\Test#1#2#3#4#5{%
  \begingroup
    \setbox0=\hbox{%
      \begingroup
        \chardef\HyColor@PdfVersion=6 %
        \HyColor@AttachfileColor{#1}\spec\inlinemacro\annot
            \TestPackageName\TestOptionName
        \edef\inline{\inlinemacro{test}}%
        \expandafter\Expect\expandafter{\spec}{#2}%
        \expandafter\Expect\expandafter{\inline}{#3}%
        \expandafter\Expect\expandafter{\annot}{#4}%
```

```
      \endgroup
      \begingroup
        \chardef\HyColor@PdfVersion=7 %
        \HyColor@AttachfileColor{#1}\spec\inlinemacro\annot
            \TestPackageName\TestOptionName
        \edef\inline{\inlinemacro{test}}%
        \expandafter\Expect\expandafter{\spec}{#2}%
        \expandafter\Expect\expandafter{\inline}{#3}%
        \expandafter\Expect\expandafter{\annot}{#5}%
      \endgroup
    }%
    \Expect*{\the\wd0}{0.0pt}%
  \endgroup
}
\newif\ifError
\def\TestError[#1]#2#3#4#5#6{%
  \begingroup
    \global\Errorfalse
    \let\OrgPackageError\PackageError
    \def\PackageError##1##2##3{%
      \edef\TestTemp{##1}%
      \ifx\TestTemp\TestPackageName
        \Expect*{\ifError too many errors\else ok\fi}{ok}%
        \Expect*{#6}*{##2}%
        \global\Errortrue
      \else
        \OrgPackageError{##1}{##2}{##3}%
      \fi
    }%
    \setbox0=\hbox{%
      \begingroup
        \chardef\HyColor@PdfVersion=#1 %
        \HyColor@AttachfileColor{#2}\spec\inlinemacro\annot
            \TestPackageName\TestOptionName
        \edef\inline{\inlinemacro{test}}%
        \expandafter\Expect\expandafter{\spec}{#3}%
        \expandafter\Expect\expandafter{\inline}{#4}%
        \expandafter\Expect\expandafter{\annot}{#5}%
      \endgroup
      \ifx\\#6\\%
      \else
        \Expect*{\ifError ok\else missing error\fi}{ok}%
      \fi
    }%
    \Expect*{\the\wd0}{0.0pt}%
  \endgroup
}
\def\NoEmptyModel{%
  Color model 'empty' is not permitted for option '\TestOptionName'%
}
\def\ModelNoXcolor#1#2{%
  Color model '#1' is not supported\MessageBreak
  without package 'xcolor' in\MessageBreak
  '\TestOptionName=[#1]{#2}'% hash-ok
}
\def\SpecNoXColor#1{%
  This color specification is not supported\MessageBreak
  without package 'xcolor' in\MessageBreak
```

```
  `test-option=#1'%
}
\begin{qstest}{AttachfileColor}{AttachfileColor}
  \Test{}{}{}{}{}%
  \Test{0.1 0.2 0.3}{[rgb]{.1,.2,.3}}{.1 .2 .3 set-rgb}%
       {/C[.1 .2 .3]}{/C[.1 .2 .3]}%
  \Test{[gray]{0.4}}{[gray]{0.4}}{.4 set-gray}%
       {/C[.4 .4 .4]}{/C[.4]}%
  \Test{[rgb]{0.3,.2,.1}}{[rgb]{0.3,.2,.1}}{.3 .2 .1 set-rgb}%
       {/C[.3 .2 .1]}{/C[.3 .2 .1]}%
  \Test{0.0 1.0 1}{[rgb]{0,1,1}}{0 1 1 set-rgb}%
       {/C[0 1 1]}{/C[0 1 1]}%
  \Test{[gray]1}{[gray]1}{1 set-gray}{/C[1 1 1]}{/C[1]}%
  \TestError[6]{[empty]{}}{}{}{}\NoEmptyModel
  \TestError[7]{[empty]{}}{}{}{}\NoEmptyModel
  \TestError[6]{[cmyk]{.1,.2,.3,.4}}{[cmyk]{.1,.2,.3,.4}}%
            {.1 .2 .3 .4 set-cmyk}{}%
            {\ModelNoXcolor{cmyk}{.1,.2,.3,.4}}%
  \TestError[7]{[cmyk]{.1,.2,.3,.4}}{[cmyk]{.1,.2,.3,.4}}%
            {.1 .2 .3 .4 set-cmyk}{/C[.1 .2 .3 .4]}{}%
  \TestError[6]{red}{red}{}{}{\SpecNoXColor{red}}%
  \TestError[7]{red}{red}{}{}{\SpecNoXColor{red}}%
\end{qstest}
\usepackage{xcolor}
\definecolor[named]{MyGreen}{rgb}{0,0.7,0}
\definecolor{mygreen}{named}{MyGreen}
\definecolor{graynine}{gray}{0.9}
\definecolor{GraySix}{Gray}{9}
\begin{qstest}{AttachfileColorX}{AttachfileColorX}
  \Test{}{}{}{}{}%
  \Test{0.1 0.2 0.3}{[rgb]{.1,.2,.3}}{.1 .2 .3 set-rgb}%
       {/C[.1 .2 .3]}{/C[.1 .2 .3]}%
  \Test{[gray]{0.4}}{[gray]{0.4}}{.4 set-gray}%
       {/C[.4 .4 .4]}{/C[.4]}%
  \Test{[rgb]{0.3,.2,.1}}{[rgb]{0.3,.2,.1}}{.3 .2 .1 set-rgb}%
       {/C[.3 .2 .1]}{/C[.3 .2 .1]}%
  \Test{0.0 1.0 1}{[rgb]{0,1,1}}{0 1 1 set-rgb}%
       {/C[0 1 1]}{/C[0 1 1]}%
  \Test{[gray]1}{[gray]1}{1 set-gray}{/C[1 1 1]}{/C[1]}%
  \Test{red}{red}{1 0 0 set-rgb}{/C[1 0 0]}{/C[1 0 0]}%
  \Test{black}{black}{0 set-gray}{/C[0 0 0]}{/C[0]}%
  \Test{cyan}{cyan}{1 0 0 0 set-cmyk}{/C[0 1 1]}{/C[1 0 0 0]}%
  \Test{[named]{black}}{[named]{black}}{0 0 0 set-rgb}%
       {/C[0 0 0]}{/C[0 0 0]}%
  \Test{[Gray]{9}}{[Gray]{9}}{.6 set-gray}{/C[.6 .6 .6]}{/C[.6]}%
  \Test{[HTML]{0080FF}}{[HTML]{0080FF}}{0 .50195 1 set-rgb}%
       {/C[0 .50195 1]}{/C[0 .50195 1]}%
  \Test{graynine}{graynine}{.9 set-gray}{/C[.9 .9 .9]}{/C[.9]}%
  \Test{GraySix}{GraySix}{.6 set-gray}{/C[.6 .6 .6]}{/C[.6]}%
  \Test{[wave]{500}}{[wave]{500}}{0 1 .49846 set-rgb}%
       {/C[0 1 .49846]}{/C[0 1 .49846]}%
  \TestError[6]{[empty]{}}{}{}{}\NoEmptyModel
  \TestError[7]{[empty]{}}{}{}{}\NoEmptyModel
\end{qstest}
⟨/test3⟩
⟨*test⟩
```

```
\begin{document}
\end{document}
⟨/test⟩
```

3.2 Test for package **xcolor**

```
⟨*test-xcolor⟩
\NeedsTeXFormat{LaTeX2e}
\nofiles
\documentclass{minimal}
⟨*xcol1⟩
\usepackage{xcolor}
\usepackage{xcolor-patch}[2016/05/16]
⟨/xcol1⟩
⟨*xcol2⟩
\usepackage{scrlfile}
\usepackage{xcolor-patch}[2016/05/16]
\usepackage{xcolor}
⟨/xcol2⟩
⟨*xcol3⟩
\usepackage{xcolor-patch}[2016/05/16]
\usepackage{xcolor}
\begin{document}
⟨/xcol3⟩
\makeatletter
\newcommand*{\ColModList}{%
  rgb,%
  cmy,%
  cmyk,%
  hsb,%
  Hsb,%
  tHsb,%
  gray,%
  RGB,%
  HTML,%
  HSB,%
  Gray,%
  % wave,
}
\newcommand*{\StartModel}{rgb}
\newcommand*{\StartValues}{.1,.2,.3}
\@for\x:=\ColModList\do{%
  \ifx\x\@empty
  \else
    \convertcolorspec\StartModel\StartValues\x\y
    \typeout{* [\StartModel]{\StartValues} ==> [\x]{\y}}%
    \@for\xx:=\ColModList\do{%
      \ifx\xx\@empty
      \else
        \convertcolorspec\x\y\xx\yy
        \typeout{* [\x]{\y} ==> [\xx]{\yy}}%
      \fi
    }%
  \fi
}
⟨xcol3⟩\end{document}
⟨xcol1 | xcol2⟩\@@end
⟨/test-xcolor⟩
```

3.2.1 Test for \@frameb@x/\fbox

```
⟨*test-xcolor-fbox⟩
\NeedsTeXFormat{LaTeX2e}
\documentclass{article}
\usepackage{xcolor}
\usepackage{xcolor-patch}[2016/05/16]
\makeatletter
\protected@edef\x{\fbox{abc}}
\let\@tempa\@undefined
\protected@edef\x{\fbox{abc}}
\makeatother
\begin{document}
\MakeUppercase{\fbox{abc}}
\end{document}
⟨/test-xcolor-fbox⟩
```

4 Installation

4.1 Download

Package. This package is available on CTAN[1]:

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
```

[1] http://ctan.org/pkg/hycolor

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

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

If you have a `docstrip.cfg` that configures and enables docstrip's TDS installing feature, then some files can already be in the right place, see the documentation of docstrip.

4.4 Refresh file name databases

If your TEX distribution (teTEX, mikTEX, ...) relies on file name databases, you must refresh these. For example, teTEX users run `texhash` or `mktexlsr`.

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 pdfLATEX:

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

5 Catalogue

The following XML file can be used as source for the TeX Catalogue. The elements `caption` and `description` are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is `hycolor.xml`.

⟨*catalogue⟩
```
<?xml version='1.0' encoding='us-ascii'?>
<!DOCTYPE entry SYSTEM 'catalogue.dtd'>
<entry datestamp='$Date$' modifier='$Author$' id='hycolor'>
  <name>hycolor</name>
  <caption>Implements colour for packages hyperref and bookmark.</caption>
  <authorref id='auth:oberdiek'/>
  <copyright owner='Heiko Oberdiek' year='2007-2011'/>
  <license type='lppl1.3'/>
  <version number='1.8'/>
  <description>
    This package provides the code for the color option
    that is used by packages <xref refid='hyperref'>hyperref</xref>
    and <xref refid='bookmark'>bookmark</xref>.
    It is not intended as package for the user.
    <p/>
    The package is part of the <xref refid='oberdiek'>oberdiek</xref> bundle.
  </description>
  <documentation details='Package documentation'
      href='ctan:/macros/latex/contrib/oberdiek/hycolor.pdf'/>
  <ctan file='true' path='/macros/latex/contrib/oberdiek/hycolor.dtx'/>
  <miktex location='oberdiek'/>
  <texlive location='oberdiek'/>
  <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'/>
</entry>
```
⟨/catalogue⟩

6 History

[2007/04/09 v1.0]

- First version.

[2007/04/11 v1.1]

- Line ends sanitized.

[2008/07/29 v1.2]

- Support for package attachfile2 added.

[2008/08/01 v1.3]

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

[2008/09/08 v1.4]

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

[2009/10/02 v1.5]

- Doku fixes (Herbert Voss).

[2009/12/12 v1.6]

- Short info shortened.

[2011/01/30 v1.7]

- Package xcolor-patch uses package hopatch.

[2016/05/16 v1.8]

- Documentation updates.

7 Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; plain numbers refer to the code lines where the entry is used.

D

E

F

H

I

L

M

N

O

P

R

S

T

U

W

X

Y

Z