The *shdoc* package – Manual for version v2.0, 2015/05/24

Simon Michael Laube
simon.laube@gmx.at

```
01  simon@linuxmint ~ $  tex
02 This is TeX, Version 3.14159265 (TeX Live 2014)
03 **\relax
04 *Hello World!
05 *\bye
```

Abstract

The *shdoc* packages helps documenting commandline actions in a fancy way. It tries to imitate the look and feel of the original shell prompt while offering a wide range of personalization options.

Contents

1 Introduction

Many people who use LaTeX and do a bit of coding in other languages have used LaTeX's *listings* package at least once. However, when me and my colleague wrote the documentation of our diploma project in 2015, we stumbled across a "problem"[1] with *listings*. A lot of command line scripting had to be done for our project to configure a Raspberry Pi B+ and those actions needed to be documented. *listings* seemed to be the wrong package to look for, so we wrote our own. At first there were few small macros that we referred to as *bashdoc* – this was version 1.0. Later the macros got a bit more complex and we threw away the *ba* part of *bashdoc*, that's how the *shdoc* package was born. The basic idea somehow stayed the same since the very first draft, but the package is much more applicable now, although there are still many things missing that could make the package better.

I hope that *shdoc* fits your application as it fitted ours. If there are any serious problems, feel free to send me an email[2]. Moreover, I do not guarantee full compatibility with other packages. The package is distributed under the LaTeX Project Public License version 1.3.

2 User commands

shdoc implements user commands that allow one to typeset the content/commands of a command line session appropriately. Further there are macros to create and modify styles and appearance of the final output.

2.1 Package inclusion and options

Just like every LaTeX package, *shdoc* can be included using the standard `\usepackage` command:

```
\usepackage[⟨options⟩]{shdoc}
```

Since version 2.0 *shdoc* supports several package options to make the environment more adjustable for users. The available options are described in detail in the following sections.

2.1.1 Color options

The shbox environment uses plenty of *different* colors, which can be adjusted through optional package arguments. Table 1 shows the adjustable colors. Every color is specified using a key-value element as an optional package argument. If no argument is used for a specific color the default value is used.

`backgroundcolor` The background color specifies the color of the background of the "shell" in the shbox environment – see section 3. The default value is set to

```
backgroundcolor=gray!70
```

to imitate the look of a shell and offer a not to dark, printable background at the same time.

[1] obviously it's not a problem, but we wanted something more colourful

[2] Please do not send spam mails of any kind. Plain text email format would be a great idea, too.

`usernamecolor` The username color specifies the color of the current user's name within the shbox environment. The default value for this color is

```
usernamecolor=green!80!yellow
```

`machinenamecolor` The machine-name color specifies the color of the machine's name within the shbox environment. The default value for this color is the same as the username color.

`indicatorcolor` The indicator separates the displayed path, user and machine values from the user input. It's default color is

```
indicatorcolor=RoyalBlue
```

`separatorcolor` The separator separates the username from the machine-name. Its default color is the same as the username color, namely

```
separatorcolor=green!80!yellow
```

Name	Description
backgroundcolor	color of the environment background
usernamecolor	color of the username
machinenamecolor	color of the machine-name
indicatorcolor	color of the indicator symbol
separatorcolor	color of the user-machine separator symbol
pathcolor	the color of the current path
optioncolor	color of the little option box
textcolor	color of the rest of the text

Table 1: Adjustable colors of the shbox environment and their description

`pathcolor` The pathcolor specifies the color of the current path and is by default the same as the indicator color:

```
pathcolor=RoyalBlue
```

`optioncolor` The option color specifies the background color of the little optionbox in the `\shoutput` command. The boxes' default color is

```
optioncolor=white!60!gray
```

`textcolor` The text color simply specifies the color of the whole rest of all text inside the shbox environment. Its default value is *black*.

2.1.2 Symbol options

In total, only two symbols can be changed in the *shdoc* package. Table 2 shows their name and description. Just as the colors, every symbol is changed using a key-value element as an optional package argument.

`indicatorsymbol` The indicator symbol separates the displayed username, machine-name and path values from the user input. Its default value is

```
indicatorsymbol=\$
```

`separatorsymbol` The separator symbol separates the username from the machine-name. Its default value is

```
separatorsymbol=@
```

Please pay attention, that the @'s *catcode* is changed to letter internally. Thus, the @ needs to be specified without a \.

Name	Description
indicatorsymbol	separates user input and displayed path
separatorsymbol	separates user- and machine-name

Table 2: Adjustable symbols of the shbox environment and their description

2.1.3 Presets

As described above, *shdoc* has some settings that can be changed by the user. Some setting values have been tested and stored as a preset that a user can easily load. Further, users are able to create their own presets via built-in commands.

`\shpreset` One can easily load a preset with `\shpreset` by using the preset's name as an argument to that macro. When a preset has been loaded it is valid for every new *shbox* environment afterwards, until the `\shpreset` macro is called again.

`\shpresetdef` Most of the time, users might want to create their own styles, which can be achieved by defining a new preset via `\shpresetdef`. The command has two mandatory arguments: The first one is the desired preset-name and the second one is the preset definition. For example one can use:

```
\shpresetdef{useless}{\relax}
```

and is then able to apply the preset by calling:

```
\shpreset{useless}
```

`\shchangecolor` To make creating presets easier there are two additional commands that simply change the according value within the package. If a user wants to change a specific color, he has to use `\shchangecolor`, where the first argument is the colors name – see section 2.1.1 – and the second one is the desired color (according to the *xcolor* syntax).

`\shchangesymbol` The same rules apply to the `\shchangesymbol` command that implements the same function for the according symbols. For further information about preset definitions, please have a look at section 4.2.

Several presets have been defined by the author and are included in the *shdoc* package. One can use them by calling the `\shpreset{`⟨*name*⟩`}` command with the name of the desired preset. The following list shows the appearance and name of the predefined presets – for visuals see section 3:

default/mint The default preset. It implements a gray background theme with light gray option boxes, light green user- and machine-name and a blue path value.

arrows Not recommended for printed documents. Implements a black background theme with gray option boxes, red username, white separator, light green machine-name, blue path and two arrows as an indicator symbol. The rest of the text is white, too.

darkblue Implements a dark blue background theme with white text, light blue option boxes, orange user- and machine-name and a green path value.

airy A light colored style. Implements a light blue background theme with cyan option boxes and user- & machine-name, as well as light orange path value and black text.

blackwhite Best for non-color prints. Implements a light gray background theme with white option boxes, black text, darker gray user- and machine-name and a dark gray path value.

2.2 Name variables

For international compatibility the *shdoc* packages implements two commands to make the list and caption name easily accessible.

`\shlistname` One can use `\shlistname` to change the title of the sessions listThe default value is "`\shlistname`{List of Terminal Sessions}".

`\shfloatname` In the same manner as described above, the float caption that's displayed below each float could be changed. Its default value is "`\shfloatname`{Terminal Session}".

2.3 The environment

Basically a complete terminal session consists out of two environments that contain the necessary information that is used to typeset the session. The outer environment is named `\sh` and basically acts as the float environment that could contain one or more so called `\shbox`es. Every `\shbox` contains in- and output lines of the terminal, with information provided by the user. The following code shows a brief overview of the possible and necessary commands:

`\sh`

`\shbox`

```
\shpreset{mint}
\begin{sh}
  \shuser{simon}
  \shmachine{linuxmint}
  \begin{shbox}
     \shline{}{cd Desktop/}
     \shline{Desktop/}{whoami}
     \shoutput{}{simon}
  \end{shbox}
\end{sh}
```

`\shuser` At least once before an occurrence of a `\shbox`, the user- and machine-name must `\shmachine` be specified. They are set globally and thus are valid for every `\shbox` that follows the macro call.

`\shline` Within the `\shbox` there are only two valid commands, namely `\shline`, which `\shoutput` is basically the users' input, and `\shoutput`, which is ment to be the machine output. `\shline` has two mandatory arguments, where the first one is the path

value[3] and the second one is the command that's about to be run. `\shoutput` implements two mandatory macros, too. Here, the first argument is an arbitrary option, which could also be left blank, when no option box is wanted. The second argument is for the commands' output.

3 Examples

3.1 Various examples

3.1.1 Basic Sessions

```
\shpreset{blackwhite}
\begin{sh}
    \shuser{simon}
    \shmachine{linuxmint}
    \begin{shbox}
        \shline{}{cd Desktop/}
        \shline{Desktop/}{xsensors -help}
        \shoutput{}{}
        \shoutput{}{\underline{Options:}}
        \shoutput{}{}
        \shoutput{-f} {Display all temperatures in Fahrenheit.}
        \shoutput{-h} {Display this help text and exit.}
        \shoutput{-c} {+filename Specify the libsensors configuration file.}
        \shoutput{-i} {+filename Specify the image file to use as a theme.}
        \shoutput{-t} {+time Specify the update time in number of seconds.}
        \shoutput{-v} {Display version number.}
    \end{shbox}
    \caption{The options of \textit{xsensors}}
    \label{sh:xsensor}
\end{sh}
```

```
01  simon@linuxmint ~ $  cd Desktop/
02  simon@linuxmint Desktop/ $ xsensors -help
03
04  Options:
05
06  [-f]     Display all temperatures in Fahrenheit.
07  [-h]     Display this help text and exit.
08  [-c]     +filename Specify the libsensors configuration file.
09  [-i]     +filename Specify the image file to use as a theme.
10  [-t]     +time Specify the update time in number of seconds.
11  [-v]     Display version number.
```

Terminal Session 1: The options of *xsensors*

[3] a smarter solution for the path value is currently in work as a draft.

```
\shpreset{default}
\begin{sh}
    \shuser{simon}
    \shmachine{linuxmint}
      \begin{shbox}
        \shline{}{tex}
            \shoutput{}{This is TeX, Version 3.14159265 (TeX Live 2014)}
            \shoutput{}{**\cmd{\relax}}
            \shoutput{}{*Hello World!}
            \shoutput{}{*\cmd{\bye}}
          \end{shbox}
    \caption{Hello World in \TeX{}}
    \label{sh:TeX}
\end{sh}
```

```
01  simon@linuxmint ~ $  tex
02 This is TeX, Version 3.14159265 (TeX Live 2014)
03 **\relax
04 *Hello World!
05 *\bye
```

Terminal Session 2: Hello World in TEX

3.1.2 Preset creation

```
\shpresetdef{odd}{
    \shpreset{blackwhite}
    \shchangecolor{usernamecolor}{orange}
    \shchangecolor{machinenamecolor}{orange}
}
\shpreset{odd}
\begin{sh}
    \shuser{doctorX}
    \shmachine{supercomputer}
    \begin{shbox}
        \shline{/home/doctorX/Documents}{exit}
    \end{shbox}
    \caption{Go away.}
    \label{sh:exit}
\end{sh}
```

```
01  doctorX@supercomputer /home/doctorX/Documents $ exit
```

Terminal Session 3: Go away.

```
\shpresetdef{mystyle}{
    \shchangecolor{backgroundcolor}{white}
    \shchangecolor{usernamecolor}{red}
    \shchangecolor{machinenamecolor}{red}
    \shchangecolor{separatorcolor}{RoyalBlue}
    \shchangecolor{optioncolor}{orange!50!yellow}
    \shchangesymbol{indicatorsymbol}{!}
    \shchangesymbol{separatorsymbol}{$\geq$}
}

\begin{sh}
    \shuser{joe}
    \shmachine{joesraspian}%
    \begin{shbox}
        \shline{}{history}
        \shoutput{480}{rubber}
        \shoutput{481}{xsensors}
        \shoutput{482}{tracepath www.google.com}
        \shoutput{483}{whoami}
        \shoutput{484}{help}
        \shoutput{485}{exit}
    \end{shbox}
    \caption{History of my session}
    \label{sh:history}
\end{sh}
```

```
01  joe≥joesraspian ~ !   history
02 [480]     rubber
03 [481]     xsensors
04 [482]     tracepath www.google.com
05 [483]     whoami
06 [484]     help
07 [485]     exit
```

Terminal Session 4: History of my session

3.2 All the predefined presets

```
01  simon@linuxmint ~ $  cd Desktop/
02  simon@linuxmint Desktop/ $ xsensors -help
03
04  Options:
05
06  [-f]     Display all temperatures in Fahrenheit.
07  [-h]     Display this help text and exit.
08  [-c]     +filename Specify the libsensors configuration file.
09  [-i]     +filename Specify the image file to use as a theme.
10  [-t]     +time Specify the update time in number of seconds.
11  [-v]     Display version number.
```

Terminal Session 5: The options of *xsensors* – *default* or *mint* preset

```
01  simon@linuxmint ~ ≫  cd Desktop/
02  simon@linuxmint Desktop/ ≫ xsensors -help
03
04  Options:
05
06  [-f]     Display all temperatures in Fahrenheit.
07  [-h]     Display this help text and exit.
08  [-c]     +filename Specify the libsensors configuration file.
09  [-i]     +filename Specify the image file to use as a theme.
10  [-t]     +time Specify the update time in number of seconds.
11  [-v]     Display version number.
```

Terminal Session 6: The options of *xsensors* – *arrows* preset

```
simon@linuxmint ~ $  cd Desktop/
simon@linuxmint Desktop/ $ xsensors -help

Options:

[-f]     Display all temperatures in Fahrenheit.
[-h]     Display this help text and exit.
[-c]     +filename Specify the libsensors configuration file.
[-i]     +filename Specify the image file to use as a theme.
[-t]     +time Specify the update time in number of seconds.
[-v]     Display version number.
```

Terminal Session 7: The options of *xsensors* – *darkblue* preset

```
simon@linuxmint ~ $  cd Desktop/
simon@linuxmint Desktop/ $ xsensors -help

Options:

[-f]     Display all temperatures in Fahrenheit.
[-h]     Display this help text and exit.
[-c]     +filename Specify the libsensors configuration file.
[-i]     +filename Specify the image file to use as a theme.
[-t]     +time Specify the update time in number of seconds.
[-v]     Display version number.
```

Terminal Session 8: The options of *xsensors* – *airy* preset

```
simon@linuxmint ~ $  cd Desktop/
simon@linuxmint Desktop/ $ xsensors -help

Options:

[-f]    Display all temperatures in Fahrenheit.
[-h]    Display this help text and exit.
[-c]    +filename Specify the libsensors configuration file.
[-i]    +filename Specify the image file to use as a theme.
[-t]    +time Specify the update time in number of seconds.
[-v]    Display version number.
```

Terminal Session 9: The options of *xsensors – blackwhite* preset

4 Implementation

This section describes the implementation of the *shdoc* package and its features. The package was written as a `*.dtx` source file and therefore the package code begins with the following instruction:

```
⟨*package⟩
```

4.1 Options

First of all, the package options for colouring and symbol options and their default values are defined. The options are then processed to make them available for the user.

color, symbol

```
\DeclareStringOption[gray!70]{backgroundcolor}
\DeclareStringOption[green!80!yellow]{usernamecolor}
\DeclareStringOption[green!80!yellow]{machinenamecolor}
\DeclareStringOption[RoyalBlue]{pathcolor}
\DeclareStringOption[RoyalBlue]{indicatorcolor}
\DeclareStringOption[green!80!yellow]{separatorcolor}
\DeclareStringOption[white!60!gray]{optioncolor}
\DeclareStringOption[black]{textcolor}
\DeclareStringOption[\$]{indicatorsymbol}
\DeclareStringOption[\~]{rootpathsymbol}
\DeclareStringOption[@]{separatorsymbol}
\ProcessKeyvalOptions*
```

4.2 Presets

Right after the option definitions the preset commands are defined. Every preset is stored in a macro with the generic name `\sh@preset@NAME`, where `NAME` stands for the preset name. To make the presets better loadable for users `\shpreset` is defined to simply execute the package internal preset macro:

\shpreset

```
\def\shpreset#1{\csname sh@preset@#1\endcsname}
```

To simplify the creation of new presets a few macros are needed. The main command is `\shpresetdef`, which creates a new preset macro with the desired settings. The macro calls `\shpreset{default}` at the beginning, so the inital settings of the new preset are the default settings of *shdoc*.

\shpresetdef

```
\def\shpresetdef#1#2{
\expandafter\gdef\csname sh@preset@#1\endcsname{\shpreset{default} #2}
}
```

Basically, every command can be used as the second argument of `\shpresetdef`, but most of them won't change any settings in *shdoc*. The only useful macros in this context are those, who explicitly change a color or symbol within the package, namely `\shchangecolor` and `\shchangesymbol`.

`\shchangecolor` has two arguments, where the first is the name of the color that should be changed and the second is the new value. Although there won't be an error, colornames that aren't listed in table 1 are not valid. The macro simply redefines the according color.

\shchangecolor

```
\def\shchangecolor#1#2{\expandafter\gdef\csname shdoc@#1\endcsname{#2}}
```

\shchangesymbol `\shchangesymbol` implements the same definition as `\shchangecolor`. Thus, the command only exists for logical reasons. It simply redefines the desired symbol.

```
\let\shchangesymbol\shchangecolor
```

Since `\shpresetdef` calls `\shpreset{default}`, the default preset needed to be defined manually via `\def`.

\sh@preset@default

```
\def\sh@preset@default{
\shchangecolor{backgroundcolor}{gray!70}
\shchangecolor{usernamecolor}{green!80!yellow}
\shchangecolor{machinenamecolor}{green!80!yellow}
\shchangecolor{pathcolor}{RoyalBlue}
\shchangecolor{indicatorcolor}{RoyalBlue}
\shchangecolor{separatorcolor}{green!80!yellow}
\shchangecolor{optioncolor}{white!60!gray}
\shchangecolor{textcolor}{black}
\shchangesymbol{indicatorsymbol}{\$}
\shchangesymbol{rootpathsymbol}{\~}
\shchangesymbol{separatorsymbol}{@}
}
```

As the *shdoc* package was initally ment to imitate the Linux Mint bash, the default preset is equal to the *mint* style, which is defined right after the default preset:

\sh@preset@mint

```
\shpresetdef{mint}{\shpreset{default}}
```

The second preset style is the “arrows” style, which is a dark color scheme with two arrows as the indicator symbol.

\sh@preset@arrows

```
\shpresetdef{arrows}{
\shchangecolor{usernamecolor}{red}
\shchangecolor{machinenamecolor}{green!80!yellow}
\shchangecolor{separatorcolor}{white}
\shchangecolor{indicatorsymbol}{$\gg$}
\shchangecolor{indicatorcolor}{green!80!yellow}
\shchangecolor{backgroundcolor}{black}
\shchangecolor{textcolor}{white}
\shchangecolor{optioncolor}{gray}
}
```

Another dark scheme is the “darkblue” preset, which defines a dark blue background color and more or less appropriate other colors. The definition of this preset is furthermore a nice example, how macros of other packages could also make sense within the `\shpresetdef` command.

\sh@preset@darkblue

```
\shpresetdef{darkblue}{
\definecolor{shdarkblue}{RGB}{7,75,138}
\shchangecolor{backgroundcolor}{shdarkblue}
\shchangecolor{textcolor}{white}
\shchangecolor{separatorcolor}{white}
\shchangecolor{usernamecolor}{orange}
\shchangecolor{machinenamecolor}{orange}
\shchangecolor{pathcolor}{green!60!black}
\shchangecolor{optioncolor}{SkyBlue!80!black}
}
```

For those, who print their documents with the black ink cartridge only there is the “blackwhite” preset. The definition sets the background and all the other colors to different versions of gray, black and white – see the examples in section 3.

\sh@preset@blackwhite

```
\shpresetdef{blackwhite}{
\shchangecolor{backgroundcolor}{gray!30}
\shchangecolor{textcolor}{black}
\shchangecolor{separatorcolor}{black}
\shchangecolor{usernamecolor}{gray}
\shchangecolor{machinenamecolor}{gray}
\shchangecolor{pathcolor}{gray!50!black}
\shchangecolor{optioncolor}{white}
\shchangecolor{indicatorcolor}{white}
}
```

The last and probably lightest preset is the “airy” preset with light blue and green elements.

\sh@preset@airy

```
\shpresetdef{airy}{
\shchangecolor{backgroundcolor}{SkyBlue!15}
```

```
\shchangecolor{usernamecolor}{Emerald}
\shchangecolor{machinenamecolor}{Emerald}
\shchangecolor{pathcolor}{orange!70}
\shchangecolor{indicatorcolor}{orange!70}
\shchangecolor{separatorcolor}{Emerald}
\shchangecolor{optioncolor}{Emerald!30}
}
```

4.3 Environment

After all the setting definitions above the actual environment for the shell commands can be defined. The whole environment consists of a float environment with caption and label and one or more inner frameboxes, which are generated with the *mdframed* package. Further there is a line number on the left side of every box. The number is defined as a standard LaTeX counter and is initially set to 0.

`\shlinenumber`

```
\newcounter{shlinenumber}
\setcounter{shlinenumber}{0}
```

Since the outer environment is a float, a list name variable is needed to make the name adjustable. There are two macros that implement this functionality: `\@shlistname` is the name variable that holds the current value of the list name. `\shlistname` is a user command, that redefines the name according to the users' input. After the definition the standard value is set.

`\shlistname`

```
\let\@shlistname\relax
\gdef\shlistname#1{\gdef\@shlistname{#1}}
\shlistname{List of Terminal Sessions}
```

4.3.1 Float environment

Now, the whole float environment is defined. The float is named `\sh` and uses the plain float style. The float name is stored in a variable, which is implemented in the same way as the list name. Afterwards the name is set. The caption is set to be at the bottom of the float and a macro for the generation of the "List of Terminal Sessions" is defined.

`\sh`

```
\newfloat{sh}{tbph}{lsh}
\restylefloat*{sh}
\floatstyle{plain}
\let\@shfloatname\relax
\gdef\shfloatname#1{\gdef\@shfloatname{#1}}
\shfloatname{Terminal~Session}
\floatname{sh}{\@shfloatname}
\captionsetup[sh]{position=bottom}
\def\listofsh{\listof{sh}{\@shlistname}}
```

4.3.2 Framebox

For each session the user and machine values can and must be set, but they could also be set at the beginning of a document to be valid for every terminal session. In the same manner as before, two name variables and their setup commands are defined. There is no default value.

`\shuser, \shmachine`

```
\makeatletter
\let\@shuser\relax
\let\@shmachine\relax
\def\shuser#1{\gdef\@shuser{#1}}
\def\shmachine#1{\gdef\@shmachine{#1}}
```

Within the framebox `\shbox` – which is defined at the very end of the source code – there are two commands: `\shline` for user inputs and `\shoutput` for shell outputs. When `\shline` is executed it prints the line number onto the left side of the box first.

`\shline`

```
\long\def\shline#1#2{
\ttfamily\noindent\scriptsize
```

If the counter is less equal then 10, a leading zero is added; otherwise the line number is directly printed.

```
\ifnum\value{shlinenumber}<10
\textcolor{\shdoc@textcolor}{0\theshlinenumber}
\else
\textcolor{\shdoc@textcolor}{\theshlinenumber}
\fi
```

After some white spaces the username is printed in the username color. It is followed by the separator and the machine name in their appropriate colors.

```
\normalsize\ \ \textcolor{\shdoc@usernamecolor}{\@shuser}\ignorespaces
\textcolor{\shdoc@separatorcolor}{\shdoc@separatorsymbol}\ignorespaces
\textcolor{\shdoc@machinenamecolor}{\@shmachine}\ignorespaces
```

Furthermore, the path argument is checked. If it's empty, the root path symbol is automatically printed, if not the given path is used. The last element that's not a command is the indicator symbol, which is set after a white space. The rest of one `\shline` contains the commands that the user has passed through the second argument.

```
\ \ifx&#1&\ignorespaces
\textcolor{\shdoc@pathcolor}{\shdoc@rootpathsymbol\ }\ignorespaces
\textcolor{\shdoc@indicatorcolor}{\shdoc@indicatorsymbol}
\else\ignorespaces
\textcolor{\shdoc@pathcolor}{#1\ }\ignorespaces
\textcolor{\shdoc@indicatorcolor}{\shdoc@indicatorsymbol}\ignorespaces
\fi\ \textcolor{\shdoc@textcolor}{#2}
\stepcounter{shlinenumber}\par
}
```

Just as before the `\shoutput` prints the line number first and checks if the value is under 10.

\shoutput

```
\long\def\shoutput#1#2{
\ttfamily\noindent\scriptsize
\ifnum\value{shlinenumber}<10
\textcolor{\shdoc@textcolor}{0\theshlinenumber}
\else
\textcolor{\shdoc@textcolor}{\theshlinenumber}
\fi
```

After that, the first argument is checked. If it's empty, the second argument is directly printed. In this case the user is able to use `\newline` to insert a linebreak. If the first argument isn't empty a tabular with two columns is generated, the first column contains a colorbox that contains the value of argument one. The second column contains the value of argument two. In this case no linebreak can be inserted.

```
\normalsize
\ifx&#1&
\textcolor{\shdoc@textcolor}{#2}\stepcounter{shlinenumber}
\else
\fcolorbox{\shdoc@optioncolor}{\shdoc@optioncolor}{\textcolor{\shdoc@textcolor}{[#1]}}
\begin{tabular}{l l}
& \textcolor{\shdoc@textcolor}{#2}\stepcounter{shlinenumber}
\end{tabular}
\fi\par
}
```

Finally, the mdframed box is defined through the environment definition command of the *mdframed* package. The frameline color is white and the default backgroundcolor is set. Further, the linecounter is set to 1, when the `\shbox` environment is started.

\shbox

```
\newmdenv[linecolor=white,backgroundcolor=\shdoc@backgroundcolor,
settings=\setcounter{shlinenumber}{1}]{shbox}

\makeatother
⟨/package⟩
```