Legacy Version of Package scrhack

Markus Kohm

2023-07-15

Contents

1 User Manual

You can find the user manual of the legacy version in the manual of scrhack.

2 Implementation of the Legacy Version of scrhack

2.1 Options

The package uses `\KOMAoptions` of scrkbase and needs at least the one of KOMA-Script v3.39:

```
⟨*package & option⟩
\RequirePackage{scrkbase}[2022-11-11]
\@ifpackagelater{scrkbase}{2023-07-14}{%
  \PackageInfo{scrhack}{KOMA-Script >= 3.42 detected.\MessageBreak
    In this case you should not use this legacy version of\MessageBreak
    scrhack, but should be able to use the current version\MessageBreak
    or even replace loading scrhack by loading really\MessageBreak
    needed third-party patches, e.g., floatbytocbasic.\MessageBreak
    But you are the boss ..}%
}{%
  \@ifpackagelater{scrkbase}{2022-11-11}{%
    \PackageWarningNoLine{scrhack}{KOMA-Script < 3.42 detected.\MessageBreak
      In this case the original scrhack of that version\MessageBreak
      should also be installed and used instead of this\MessageBreak
      legacy package}%
  }{%
    \PackageError{scrhack}{incompatible KOMA-Script < v3.39 detected}{%
      This error is fatal!\MessageBreak
      Loading of scrhack will be aborted!}%
    \expandafter\let\csname ver@scrhack.sty\endcsname\undefined
    \endinput
  }%
}
⟨/package & option⟩
```

The options allow the selection of the used manipulations of other packages. But all these options are valid only until both scrhack and the other package has been loaded. Later settings are invalid.

2.2 Used Macros and Commands

```
⟨*package & body⟩
```

`\scr@ifexpected` Mentioned to do (re)definitions only, if local execution of code does not change a command. Concrete, if (local) execution of {⟨*test code*⟩} of

`\scr@ifexpected{⟨command⟩}{⟨test code⟩}{⟨then code⟩}{⟨else code⟩}`

would not result in a change of \⟨*command*⟩, the {⟨*then code*⟩} will be executed, otherwise the {⟨*else code*⟩}.

Note: ⟨*test code*⟩ must not have any global side effect!

```
\newcommand{\scr@ifexpected}[2]{%
  \begingroup
    \let\@tempa#1
    #2
    \ifx\@tempa#1
      \aftergroup\@firstoftwo
    \else
      \aftergroup\@secondoftwo
    \fi
  \endgroup
}
```

`\scr@hack@load` Conditionally loading of a hack file.

`\scr@hack@load{⟨extension⟩}{⟨basename⟩}{⟨hack⟩} {⟨true code⟩}{⟨false code⟩}`

The first two arguments `{`⟨⟩`}` and `{`⟨*extension*⟩`}` and `{`⟨*basename*⟩`}` are same to the internalLaTeX kernel command `\@ifl@aded`. With LaTeX from 2021/11/15 we even could use `\@ifl@aded` for the test, because it is not restricted to the preamble any more. But because it is restricted with older LaTeX kernels, we test the definition of `\ver@`⟨*basename*⟩`.`⟨*extension*⟩. If it is defined, we also test the definition of `\ver@`⟨*basename*⟩`.hak`. If this is not defined, ⟨*hack*⟩`.hak` is loaded and ⟨*true code*⟩ is executed. Otherwise ⟨*false code*⟩ is executed. This means: A hack file is only loaded, if a class or package (or other file with version) has already been loaded, but the hack file has not already been loaded.

```
\newcommand*{\scr@hack@load}[2]{%
  \expandafter\ifx\csname ver@#2.#1\endcsname\relax
    \expandafter\@secondoftwo
  \else
```

Load a hack only once.

```
    \expandafter\ifx\csname ver@#2.hak\endcsname\relax
      \expandafter\expandafter\expandafter\@firstoftwo
    \else
      \expandafter\expandafter\expandafter\@secondoftwo
    \fi
  \fi
  {%
    \PackageInfo{scrhack}{loading #2 hack}%
    \edef\reserved@a{%
      \noexpand\makeatletter\noexpand\input{#2.hak}%
      \noexpand\catcode`\noexpand\@\the\catcode`\@\relax
    }\reserved@a
  }{%
    \PackageInfo{scrhack}{ignorring #2 hack}%
  }%
}
```

```
⟨/package & body⟩
```

2.3 The hyperref hack

Package hyperref places the anchor of the star variant of a heading after the heading, but the anchor of the normal variant before the heading even if the heading is not numbered because of the value of `secnumdepth`. The hack changes this to more consistent placing always before the heading.

`hyperref` (*opt.*)

```
⟨*package & option⟩
\KOMA@ifkey{hyperref}{@scrhack@hyperref}%
\KOMAExecuteOptions{hyperref=true}%
⟨/package & option⟩
```

This is a little bit tricky, because hyperref loads the driver file via `\AtEndOfPackage` and the hack cannot be loaded before loading the driver file. Using `\AfterPackage*` would not be good enough therefore, because it would load the hack before the driver file. But using `\AtEndOfPackage` inside `\AfterPackage*` should delay the loading enough. This also works, if hyperref has already been loaded. In this case the code will be executed at the end of scrhack instead of hyperref. This is no problem.

```
⟨*package & body⟩
\BeforePackage{hyperref}{%
  \scr@ifundefinedorrelax{hy@insteadofrefstepcounter}{}{%
    \PackageInfo{scrhack}{hyperref hack deactivated because of\MessageBreak
      detection of KOMA-Script class, that doesn't\MessageBreak
      need that hack,}%
    \KOMAExecuteOptions[scrhack.sty]{hyperref=false}%
  }%
}
\AfterPackage*{hyperref}{%
  \if@scrhack@hyperref
    \@ifpackagelater{hyperref}{2009/11/24}{%
      \PackageInfo{scrhack}{hyperref hack deactivated because of\MessageBreak
        detection of hyperref version, that doesn't\MessageBreak
        need that hack,}%
      \KOMAExecuteOptions[scrhack.sty]{hyperref=false}%
    }{%
      \AtEndOfPackage{%
        \KOMA@key[scrhack.sty]{hyperref}[true]{%
          \PackageWarning{scrhack}{option `hyperref=#1' ignored}%
          \FamilyKeyStateProcessed
        }%
        \if@scrhack@hyperref\scr@hack@load\@pkgextension{hyperref}\fi
      }%
    }%
  \fi
}
⟨/package & body⟩
```

`\@schapter`
`\@spart`
`\@ssect`

We do not hack hyperref itself, but the driver files. After loading hyperref the driver file is also loaded. Moreover, all driver files that re-define these commands have to be patched also. So the patch is related to the loading of hyperref instead of every single driver file. This have already been tested by `\scr@hack@load`. So we just have to test, whether the definition of these commands are the expected one. derzeit den erwarteten Inhalt haben.

```
⟨*hyperref & body⟩
\scr@ifexpected\@schapter{%
  \def\@schapter#1{%
    \H@old@schapter{#1}%
    \begingroup
      \let\@mkboth\@gobbletwo
      \Hy@GlobalStepCount\Hy@linkcounter
      \xdef\@currentHref{\Hy@chapapp*.\the\Hy@linkcounter}%
      \Hy@raisedlink{%
        \hyper@anchorstart{\@currentHref}\hyper@anchorend
      }%
    \endgroup
  }%
}{%
  \PackageInfo{scrhack}{redefining \string\@schapter}%
  \def\@schapter#1{%
    \begingroup
      \let\@mkboth\@gobbletwo
      \Hy@GlobalStepCount\Hy@linkcounter
      \xdef\@currentHref{\Hy@chapapp*.\the\Hy@linkcounter}%
      \Hy@raisedlink{%
        \hyper@anchorstart{\@currentHref}\hyper@anchorend
      }%
    \endgroup
    \H@old@schapter{#1}%
  }%
}{%
  \scr@ifexpected\@schapter{%
    \def\@schapter#1{%
      \begingroup
        \let\@mkboth\@gobbletwo
        \Hy@GlobalStepCount\Hy@linkcounter
        \xdef\@currentHref{\Hy@chapapp*.\the\Hy@linkcounter}%
        \Hy@raisedlink{%
          \hyper@anchorstart{\@currentHref}\hyper@anchorend
        }%
      \endgroup
      \H@old@schapter{#1}%
    }%
  }{}{%
    \PackageWarningNoLine{scrhack}{unknown \string\@schapter\space
      definition found!\MessageBreak
      Maybe you are using a unsupported hyperref version}%
  }%
}
\scr@ifexpected\@spart{%
  \def\@spart#1{%
    \H@old@spart{#1}%
    \Hy@GlobalStepCount\Hy@linkcounter
    \xdef\@currentHref{part*.\the\Hy@linkcounter}%
    \Hy@raisedlink{%
      \hyper@anchorstart{\@currentHref}\hyper@anchorend
    }%
  }%
}{%
  \PackageInfo{scrhack}{redefining \string\@spart}%
  \def\@spart#1{%
    \Hy@GlobalStepCount\Hy@linkcounter
    \xdef\@currentHref{part*.\the\Hy@linkcounter}%
    \Hy@raisedlink{%
      \hyper@anchorstart{\@currentHref}\hyper@anchorend
    }%
    \H@old@spart{#1}%
  }%
}{%
  \scr@ifexpected\@spart{%
    \def\@spart#1{%
```

```
   \Hy@GlobalStepCount\Hy@linkcounter
   \xdef\@currentHref{part*.\the\Hy@linkcounter}%
   \Hy@raisedlink{%
    \hyper@anchorstart{\@currentHref}\hyper@anchorend
   }%
   \H@old@spart{#1}%
  }%
 }{}{%
  \PackageWarningNoLine{scrhack}{unknown \string\@spart\space
   definition found!\MessageBreak
   Maybe you are using a unsupported hyperref version}%
 }%
}
\scr@ifexpected\@ssect{%
 \def\@ssect#1#2#3#4#5{%
  \H@old@ssect{#1}{#2}{#3}{#4}{#5}%
  \phantomsection
 }%
}{%
 \PackageInfo{scrhack}{redefining \string\@ssect}%
 \def\@ssect#1#2#3#4#5{%
  \H@old@ssect{#1}{#2}{#3}{#4}{\phantomsection\ignorespaces#5}%
 }%
}{%
 \scr@ifexpected\@ssect{%
  \def\@ssect#1#2#3#4#5{%
   \H@old@ssect{#1}{#2}{#3}{#4}{\phantomsection\ignorespaces#5}%
  }%
 }{}{%
  \PackageWarningNoLine{scrhack}{unknown \string\@ssect\space
   definition found!\MessageBreak
   Maybe you are using a unsupported hyperref version}%
 }%
}
⟨/hyperref & body⟩
```

2.4 The floatrow hack

Package floatrow insists in using `\float@listhead` for the setting of the headings. This is not recommended any longer since KOMA-Script 2.95 from July 2006! The KOMA-Script classes will remove the support for this very limited approach. Recommended is the usage of tocbasic. The effort to do so, it very small and I do not understand, why package authors are not willing to use it. Because the benefit would be a lot.

This hack adds tocbasic usage to floatrow. Nevertheless it is more complicated than adding tocbasic support into floatrow.

`floatrow` (*opt.*)

```
⟨*package & option⟩
\KOMA@ifkey{floatrow}{@scrhack@floatrow}
\KOMAExecuteOptions{floatrow=true}
⟨/package & option⟩
```

```
⟨*package & body⟩
\AfterPackage*{floatrow}{%
  \KOMAoptions{float=false}%
  \PackageInfo{scrhack}{option ‘float’ deactivated}%
  \KOMA@key[scrhack.sty]{floatrow}{%
    \PackageWarning{scrhack}{option ‘floatrow’ ignored}%
    \FamilyKeyStateProcessed
  }%
  \if@scrhack@floatrow\scr@hack@load\@pkgextension{floatrow}\fi
}
⟨/package & body⟩
```

`\DeclareNewFloatType` `\listof` Command `\DeclareNewFloatType` is used to define a new float environment. We have to add registration of the file extension (third argument) with tocbasic. Command `\listof` outputs the list of the float environments. Here we just use the corresponding macro of tocbasic.

`\float@addtolists` The deprecated macro `\float@addtolists` is not needed any longer an will be reset to the original definition.

```
⟨*floatrow & body⟩
\scr@ifexpected{\DeclareNewFloatType}{%
  \long\def\DeclareNewFloatType#1#2{\def\FB@captype{#1}%
    \expandafter\edef\csname ftype@#1\endcsname{\the\c@float@type}%
    \addtocounter{float@type}{\value{float@type}}%
    \@namedef{#1name}{#1}\newcounter{#1}%
    \expandafter\edef\csname fnum@#1\endcsname
    {\expandafter\noexpand\csname #1name\endcsname\nobreakspace
      \expandafter\noexpand\csname the#1\endcsname}%
    \@namedef{the#1}{\arabic{#1}}\flnew@ext{lo#1}\@namedef{fps@#1}{tbp}%
    \@namedef{l@#1}{\@dottedtocline{1}{1.5em}{2.3em}}%
    \caption@setkeys[floatrow]{newfloat}{#2}\let\FR@tmp=\relax
    \xdef\@tempa{\noexpand\flrow@types{\the\flrow@types \FR@tmp{#1}}}%
    \@tempa}%
}{%
  \scr@ifexpected{\listof}{%
    \def\listof#1#2{%
      \@ifundefined{ext@#1}{\flrow@error{Unknown float style ‘#1’}}{%
        \expandafter\providecommand\csname l@#1\endcsname
        {\@dottedtocline{1}{1.5em}{2.3em}}%
        \float@listhead{#2}%
        \begingroup\setlength{\parskip}{\z@}%
        \@starttoc{\@nameuse{ext@#1}}%
        \endgroup}}%
  }{%
    \RequirePackage{tocbasic}%
    \PackageInfo{scrhack}{redefining \string\DeclareNewFloatType}%
```

Note: It would be better to first test, if the new extension is already in use like done at the float hack. But I don't know the floatrow code good enough to make such a change!

```
    \renewcommand\DeclareNewFloatType[2]{\def\FB@captype{#1}%
```

```
      \expandafter\edef\csname ftype@#1\endcsname{\the\c@float@type}%
      \addtocounter{float@type}{\value{float@type}}%
      \@namedef{#1name}{#1}\newcounter{#1}%
      \expandafter\edef\csname fnum@#1\endcsname
      {\expandafter\noexpand\csname #1name\endcsname\nobreakspace
        \expandafter\noexpand\csname the#1\endcsname}%
      \@namedef{the#1}{\arabic{#1}}\flnew@ext{lo#1}\@namedef{fps@#1}{tbp}%
      \DeclareTOCStyleEntry[level=1,numwidth=2.3em,indent=1.5em]{default}{#1}%
      \caption@setkeys[floatrow]{newfloat}{#2}\let\FR@tmp=\relax
      \xdef\@tempa{\noexpand\flrow@types{\the\flrow@types \FR@tmp{#1}}}%
      \@tempa
      \xdef\@tempa{\noexpand\addtotoclist[float]{\@nameuse{ext@\FB@captype}}%
        \noexpand\setuptoc{\@nameuse{ext@\FB@captype}}{chapteratlist}%
      }%
      \@tempa
    }%
    \PackageInfo{scrhack}{redefining \string\listof}%
    \renewcommand*\listof[2]{%
      \@ifundefined{ext@#1}{\flrow@error{Unknown float style ‘#1’}}{%
        \@ifundefined{l@#1}{\expandafter\let\csname l@#1\endcsname\l@figure
          \@ifundefined{l@#1}{%
            \DeclareTOCStyleEntry[level=1,numwidth=2.3em,indent=1.5em]
            {default}{#1}%
          }{}%
        }{}%
        \listoftoc[{#2}]{\csname ext@#1\endcsname}%
      }%
    }%
    \scr@ifexpected{\float@addtolists}{%
      \long\def\float@addtolists#1{%
        \def\float@do##1{\addtocontents{##1}{#1}} \the\float@exts}%
    }{%
      \PackageInfo{scrhack}{undefining \string\float@addtolists}%
      \let\float@addtolists\relax
    }{%
      \PackageWarningNoLine{scrhack}{unkown \string\float@addtolists\space
        definition found!\MessageBreak
        Maybe you are using a unsupported floatrow version}%
    }%
  }{%
    \PackageWarningNoLine{scrhack}{unknown \string\listof\space
      definition found!\MessageBreak
      Maybe you are using a unsupported floatrow version}%
  }%
}{%
  \PackageWarningNoLine{scrhack}{unknown \string\DeclareNewFloatType\space
    definition found!\MessageBreak
    Maybe you are using a unsupported floatrow version}%
}
⟨/floatrow & body⟩
```

2.5 The float hack

Package float insists in using `\float@listhead` for the setting of the headings. This is not recommended any longer since KOMA-Script 2.95 from July 2006! The KOMA-Script classes will remove the support for this very limited approach. Recommended is the usage of tocbasic. The effort to do so, it very small and I do not understand, why package authors are not willing to use it. Because the benefit would be a lot.

This hack adds tocbasic usage to float. Nevertheless it is more complicated than adding tocbasic support into float.

`float` (*opt.*)

```
⟨*package & option⟩
\KOMA@ifkey{float}{@scrhack@float}%
\KOMAExecuteOptions{float=true}%
⟨/package & option⟩
⟨*package & body⟩
\AfterPackage*{float}{%
  \KOMA@key[scrhack.sty]{float}{%
    \PackageWarning{scrhack}{option ‘float’ ignored}%
    \FamilyKeyStateProcessed
  }%
  \if@scrhack@float\scr@hack@load\@pkgextension{float}\fi
}
⟨/package & body⟩
```

`\newfloat` We have to add registration of the file extension (third argument of `\newfloat`) with tocbasic.

`\listof` Command `\listof` outputs the list of the float environments. Here we just use the corresponding macro of tocbasic.

`\float@addtolists` The deprecated macro `\float@addtolists` is not needed any longer an will be reset to the original definition.

```
⟨*float & body⟩
\scr@ifexpected{\newfloat}{%
  \long\def\newfloat#1#2#3{\@namedef{ext@#1}{#3}
    \let\float@do=\relax
    \xdef\@tempa{\noexpand\float@exts{\the\float@exts \float@do{#3}}}%
    \@tempa
    \floatplacement{#1}{#2}%
    \@ifundefined{fname@#1}{\floatname{#1}{#1}}{}
    \expandafter\edef\csname ftype@#1\endcsname{\value{float@type}}%
    \addtocounter{float@type}{\value{float@type}}
    \restylefloat{#1}%
    \expandafter\edef\csname fnum@#1\endcsname%
    {\expandafter\noexpand\csname fname@#1\endcsname{}
      \expandafter\noexpand\csname the#1\endcsname}
    \@ifnextchar[%]
    {\float@newx{#1}}%
    {\@ifundefined{c@#1}{\newcounter{#1}\@namedef{the#1}{\arabic{#1}}}%
      {}}}%
```

```
}{%
  \scr@ifexpected{\listof}{%
    \def\listof#1#2{%
      \@ifundefined{ext@#1}{\float@error{#1}}{%
        \@namedef{l@#1}{\@dottedtocline{1}{1.5em}{2.3em}}%
        \float@listhead{#2}%
        \begingroup\setlength{\parskip}{\z@}%
        \@starttoc{\@nameuse{ext@#1}}%
        \endgroup}}%
  }{%
    \RequirePackage{tocbasic}%
    \PackageInfo{scrhack}{redefining \string\newfloat}%
    \renewcommand\newfloat[3]{%
      \Ifattoclist{#3}{%
        \PackageError{scrhack}{extension ‘#3’ already in use}{%
          Each extension may be used only once.\MessageBreak
          You, the class, or another package already uses extension
          ‘#3’.\MessageBreak
          \string\newfloat\space command will be ignored!}%
      }{%
        \addtotoclist[float]{#3}%
        \setuptoc{#3}{chapteratlist}%
        \@namedef{ext@#1}{#3}%
        \let\float@do=\relax
        \xdef\@tempa{\noexpand\float@exts{\the\float@exts \float@do{#3}}}%
        \@tempa
        \floatplacement{#1}{#2}%
        \@ifundefined{fname@#1}{\floatname{#1}{#1}}{}%
        \expandafter\edef\csname ftype@#1\endcsname{\value{float@type}}%
        \addtocounter{float@type}{\value{float@type}}
        \restylefloat{#1}%
        \expandafter\edef\csname fnum@#1\endcsname%
        {\expandafter\noexpand\csname fname@#1\endcsname{}
          \expandafter\noexpand\csname the#1\endcsname}%
        \@ifnextchar[%]
        {\float@newx{#1}}%
        {\@ifundefined{c@#1}{\newcounter{#1}\@namedef{the#1}{\arabic{#1}}}%
          {}}}%
    }%
    \PackageInfo{scrhack}{redefining \string\listof}%
    \renewcommand*\listof[2]{%
      \@ifundefined{ext@#1}{\float@error{#1}}{%
        \@ifundefined{l@#1}{\expandafter\let\csname l@#1\endcsname\l@figure
          \@ifundefined{l@#1}{%
            \DeclareTOCStyleEntry[level=1,numwidth=2.3em,indent=1.5em]
            {default}{#1}%
          }{}%
        }{}%
        \listoftoc[{#2}]{\csname ext@#1\endcsname}%
      }%
    }%
    \scr@ifexpected{\float@addtolists}{%
```

```
   \long\def\float@addtolists#1{%
    \def\float@do##1{\addtocontents{##1}{#1}} \the\float@exts}%
  }{%
   \PackageInfo{scrhack}{undefining \string\float@addtolists}%
   \let\float@addtolists\relax
  }{%
   \PackageWarningNoLine{scrhack}{unkown \string\float@addtolists\space
    definition found!\MessageBreak
    Maybe you are using a unsupported float version}%
  }%
 }{%
  \PackageWarningNoLine{scrhack}{unknown \string\listof\space
   definition found!\MessageBreak
   Maybe you are using a unsupported float version}%
 }%
}{%
 \PackageWarningNoLine{scrhack}{unknown \string\newfloat\space
  definition found!\MessageBreak
  Maybe you are using a unsupported float version}%
}
⟨/float & body⟩
```

2.6 The listings hack

Package listing insists in using `\float@listhead` for the setting of the headings. This is not recommended any longer since KOMA-Script 2.95 from July 2006! The KOMA-Script classes will remove the support for this very limited approach. Recommended is the usage of tocbasic. The effort to do so, it very small and I do not understand, why package authors are not willing to use it. Because the benefit would be a lot.

This hack adds tocbasic usage to old versions of listings. Nevertheless it is more complicated than adding tocbasic support into listings.

`listings` (*opt.*) Note: The deactivation of the hack for new version of listings cannot be done comparing the listings version, because listings version 1.10 uses `\AtEndOfPackage` for `\ProvidesPackage` and therefore the version is not defined when the after package hook is executed. So we test instead the lstmisc version.

```
⟨*package & option⟩
\KOMA@ifkey{listings}{@scrhack@listings}
\KOMAExecuteOptions{listings=true}
⟨/package & option⟩
⟨*package & body⟩
\AfterPackage*{listings}{%
 \@ifpackagelater{lstmisc}{2024/02/15}{%
  \PackageInfo{scrhack}{listings hack deactivated because of\MessageBreak
   detection of listings version, that doesn't\MessageBreak
   need that hack,}%
  \KOMAExecuteOptions[scrhack.sty]{listings=false}%
 }{%
  \if@scrhack@listings\scr@hack@load\@pkgextension{listings}\fi
 }%
 \KOMA@key[scrhack.sty]{listings}[true]{%
```

```
    \PackageWarning{scrhack}{option ‘listings=#1’ ignored}%
    \FamilyKeyStateProcessed
  }%
}
⟨/package & body⟩
```

`\scr@do@hack@listings`
`\lstlistoflistings`

`\scr@do@hack@listings` patches listings to use tocbasic for the whole output of the list of listings. This adds several new features to listings.

`\ext@lstlisting`
`\lst@MakeCaption`
`\float@addtolists`

Macro `\ext@lstlisting` is defined, to make `\captionof` work with argument `lstlisting`. Additionally `\lst@MakeCaption` is patched to use the new macro. The deprecated macro `\float@addtolists` is not needed any longer an will be reset to the original definition. But, because listings delays the redefinition with `\AtBeginDocument` we have to do the same here.

```
⟨*listings & body⟩
\newcommand*{\scr@do@hack@listings}{%
  \RequirePackage{tocbasic}%
  \addtotoclist[float]{lol}%
  \setuptoc{lol}{chapteratlist}%
  \PackageInfo{scrhack}{redefining \string\lstlistoflistings}%
  \renewcommand*{\lstlistoflistings}{\listoftoc[{\lstlistlistingname}]{lol}}%
  \providecommand*{\ext@lstlisting}{lol}%
  \@tempswafalse
  \xpatchcmd\lst@MakeCaption
    {\addcontentsline{lol}}%
    {\addcontentsline{\ext@lstlisting}}%
    {\PackageInfo{scrhack}{patching \string\lst@MakeCaption}\@tempswatrue}%
    {\PackageWarning{scrhack}{cannot patch \string\lst@MakeCaption}}%
  \@whilesw\if@tempswa\fi{%
    \xpatchcmd\lst@MakeCaption
      {\addcontentsline{lol}}%
      {\addcontentsline{\ext@lstlisting}}%
      {}%
      {\@tempswafalse}%
  }%
  \AtBeginDocument{%
    \scr@ifexpected{\float@addtolists}{%
      \def\float@addtolists##1{\addtocontents{lol}{##1}}%
    }{%
      \PackageInfo{scrhack}{undefining \string\float@addtolists}%
      \let\float@addtolists\relax
    }{%
      \scr@ifexpected{\float@addtolists}{%
        \def\float@addtolists##1{\addtocontents{lol}{##1}%
          \orig@float@addtolists{##1}}%
      }{%
        \PackageInfo{scrhack}{setting \string\float@addtolists\MessageBreak
          to \string\orig@float@addtolists}%
        \let\float@addtolists\orig@float@addtolists
      }{%
        \PackageWarningNoLine{scrhack}{unkown \string\float@addtolists\space
          definition found!\MessageBreak
          Maybe you are using a unsupported listings version}%
      }%
    }%
  }%
```

`\lst@makecaption` From version 3.30 setting of `\@captype` to `lstlisting` is added to `\lst@makecaption`. This adds support for usage of `\raggedlstlistingcaption` (if the user or a package author defines it). The change is a little bit dangerous, because it will break, if listings would add an argument to `\lst@makecaption`.

```
  \expandafter\def\expandafter\lst@makecaption\expandafter{%
    \expandafter\def\expandafter\@captype\expandafter{%
      \expandafter l\expandafter s\expandafter t\expandafter l%
      \expandafter i\expandafter s\expandafter t\expandafter i%
      \expandafter n\expandafter g\expandafter}\lst@makecaption
  }%
  \let\scr@do@hack@listings\relax
}
\scr@ifexpected{\lstlistoflistings}{%
  \def\lstlistoflistings{\bgroup
    \let\contentsname\lstlistlistingname
    \let\lst@temp\@starttoc \def\@starttoc##1{\lst@temp{lol}}%
    \tableofcontents \egroup}%
}{%
  \scr@do@hack@listings
}{%
  \scr@ifexpected{\lstlistoflistings}{%
    \def\lstlistoflistings{%
      \begingroup
        \@ifundefined{@restonecoltrue}{}{%
          \if@twocolumn
            \@restonecoltrue\onecolumn
          \else
            \@restonecolfalse
          \fi
        }%
        \float@listhead{\lstlistlistingname}%
        \parskip\z@\parindent\z@\parfillskip \z@ \@plus 1fil%
        \@starttoc{lol}%
        \@ifundefined{@restonecoltrue}{}{%
          \if@restonecol\twocolumn\fi
        }%
      \endgroup
    }%
  }{%
    \scr@do@hack@listings
  }{%
    \PackageWarningNoLine{scrhack}{unknown \string\lstlistoflistings\space
      definition found!\MessageBreak
      Maybe you are using a unsupported listings version}%
  }%
}
⟨/listings & body⟩
```

2.7 The nomencl hack

With package nomencl and the KOMA-Script classes there is only a small disharmony between relating the ToC entry. Nevertheless the package can benefit from using tocbasic.

This hack adds tocbasic usage to nomencl. This is done using xpatch Nevertheless it is much more complicated than adding real tocbasic support into nomencl. The mess is, that every new release of nomencl changes the kind of "support" of tocbasic. For some versions there isn't any such support. Other versions use tocbasic if it has been loaded. Other versions need an option to activate usage of tocbasic. So don't expect that the following hack does the correct thing for every version of nomencl!

`nomencl` (*opt.*)

```
⟨*package & option⟩
\RequirePackage{xpatch}%
\KOMA@ifkey{nomencl}{@scrhack@nomencl}
\KOMAExecuteOptions{nomencl=true}
⟨/package & option⟩
⟨*package & body⟩
\AfterPackage*{nomencl}{%
  \KOMA@key[scrhack.sty]{nomencl}{%
    \PackageWarning{scrhack}{option ‘nomencl’ ignored}%
    \FamilyKeyStateProcessed
  }%
  \if@scrhack@nomencl\scr@hack@load\@pkgextension{nomencl}\fi
}
⟨/package & body⟩
```

`\thenomenclature` Instead of using the several definitions by cases of the several versions of nomencl we use a patch, that should remove all such cases and use tocbasic always.

```
⟨*nomencl & body⟩
\RequirePackage{tocbasic}
```

With nomencl ¿ 5.0 this should be all. Unfortunately is isn't, because newer versions of nomencl need option `tocbasic` to be set. Good luck: It is the default. So the chances are good, that with new versions of nomencl usage of tocbasic is the default.

```
\@ifpackagelater{nomencl}{2019/01/01}{%
  \PackageInfo{scrhack}{only ‘tocbasic’ loaded for ‘nomencl’}%
  \expandafter\endinput
}{}
```

Otherwise we have to patch it.

```
\addtotoclist[nomencl]{nlo}
\addtotoclist[nomencl]{nls}
\xpatchcmd{\thenomenclature}{%
  \@ifundefined{chapter}%
  {
    \section*{\nomname}
    \if@intoc\addcontentsline{toc}{section}{\nomname}\fi%
  }%
  {
    \chapter*{\nomname}
    \if@intoc\addcontentsline{toc}{chapter}{\nomname}\fi%
  }%
}{%
  \scr@hack@nomencl@head
}{%
  \PackageInfo{scrhack}{\string\thenomenclature\space successfully patched}%
}{%
```

From nomencl 5.0 another code is used if the nomenclature is a chapter. So we need another patch:

```
  \xpatchcmd{\thenomenclature}{%
    \@ifundefined{chapter}%
    {
      \section*{\nomname}
      \if@intoc\addcontentsline{toc}{section}{\nomname}\fi%
    }%
    {
      \chapter*{\nomname}
      \markboth{\nomname}{\nomname}%
      \if@intoc\addcontentsline{toc}{chapter}{\nomname}\fi%
    }%
  }{%
    \scr@hack@nomencl@head
  }{%
    \PackageInfo{scrhack}{\string\thenomenclature\space successfully patched}%
  }{%
```

But this code is also wrong, because it breaks manual running heads, e.g., of page style `myheadings`. So I hope, there will be another code, that needs a somehow different patch:

```
    \xpatchcmd{\thenomenclature}{%
      \@ifundefined{chapter}%
      {
        \section*{\nomname}
        \if@intoc\addcontentsline{toc}{section}{\nomname}\fi%
      }%
      {
        \chapter*{\nomname}
        \@mkboth{\nomname}{\nomname}%
        \if@intoc\addcontentsline{toc}{chapter}{\nomname}\fi%
      }%
    }{%
      \scr@hack@nomencl@head
    }{%
      \PackageInfo{scrhack}{\string\thenomenclature\space successfully patched}%
    }{%
```

If this also does not work, we try an emergency patch, that could break things.

```
      \xpatchcmd{\thenomenclature}{%
        \@ifundefined{chapter}%
      }{%
        \scr@hack@nomencl@head
        \@gobbletwo
      }{%
```

```
        \PackageWarning{scrhack}{%
          using emergency patch of \string\thenomenclature.\MessageBreak
          This could happen if you are using an unknown\MessageBreak
          release of package ‘nomencl’.\MessageBreak
          Mostly this patch does work too, but it also could\MessageBreak
          break ‘nomencl‘. If this happens, please deactivate\MessageBreak
          the ‘nomencl‘ hack of ‘scrhack‘%
        }%
      }{%
        \PackageWarning{scrhack}{%
          cannot patch \string\thenomenclature.\MessageBreak
          This could happen if you are using an unknown\MessageBreak
          release of package ‘nomencl’%
        }%
      }%
    }%
  }%
}
⟨/nomencl & body⟩
```

`\scr@hack@nomencl@head` We don’t need an extra group, because `thenomenclature` already adds the group.

```
⟨*nomencl & body⟩
\newcommand*{\scr@hack@nomencl@head}{%
  \providecommand*{\listofnlsname}{\nomname}%
  \def\@currext{nls}%
  \tocbasic@listhead{\nomname}%
}
⟨/nomencl & body⟩
```

Note: Only the heading hooks of tocbasic are used. The other hooks can be set, but would never be used!

`\if@intoc` Generally we have to support nomencl’s option `intoc` and the switch `\if@intoc`. This is done by (un-)setting tocbasic feature `totoc`. So additional changes of the feature can be done at first after loading both, nomencl and scrhack. Moreover, later changes of `\if@intoc` are not recognized.

```
⟨*nomencl & body⟩
\if@intoc
  \setuptoc{nls}{totoc}
\fi
⟨/nomencl & body⟩
```

2.8 The setspace hack

Package setspace internally uses `\@ptsize` in a way, that only works as long as `\@ptsize` is an integer. But because KOMA-Script supports non integer font sizes, `\@ptsize` can also contain non-integer values. Moreover for `11pt`, LaTeX does not use a 11 pt font but a 10.95 pt font with a `\baselineskip` of 13.6 pt, the correct factor for, e.g., `\onehalfspacing` would be:

$$10{,}95\,\mathrm{pt}\,\dot{\imath}, 5/13{,}6\,\mathrm{pt} \equiv 1{,}208$$

But setspace uses 1.213, which would be correct for 11 pt. Moreover the factor is always relative to the base font size and not the current font size. Because of this

```
\documentclass[10pt]{article}
\usepackage{setspace}
\begin{document}
\large\onehalfspacing\raggedright
Fontsize: \csname f@size\endcsname pt\\
Normal baselineskip: \csname f@baselineskip\endcsname\\
baselineskip: \the\baselineskip
\end{document}
```

results in another `\baselineskip` than:

```
\documentclass[11pt]{article}
\usepackage{setspace}
\begin{document}
\large\onehalfspacing\raggedright
Fontsize: \csname f@size\endcsname pt\\
Normal baselineskip: \csname f@baselineskip\endcsname\\
baselineskip: \the\baselineskip
\end{document}
```

despite both examples use the same font size. So, to make it absolutely correct, the factor would have to change with every change of the font size. But the hack also does not do so. It just calculates the factor depending on the base font size an the current font size. This is enough to have the same `\baselineskip` with both of the examples above.

`setspace` (*opt.*)

```
⟨*package & option⟩
\KOMA@ifkey{setspace}{@scrhack@setspace}
\KOMAExecuteOptions{setspace=true}
⟨/package & option⟩
⟨*package & body⟩
\AfterPackage*{setspace}{%
  \KOMA@key[scrhack.sty]{setspace}{%
    \PackageWarning{scrhack}{option ‘setspace’ ignored}%
    \FamilyKeyStateProcessed
  }%
  \if@scrhack@setspace\scr@hack@load\@pkgextension{setspace}\fi
}
⟨/package & body⟩
```

`\onehalfspacing` This selects the one half line spacing. I think, a factor with three digits for the decimals should be enough.

```
⟨*setspace & body⟩
\scr@ifexpected{\onehalfspacing}{%
  \long\def\onehalfspacing{%
    \setstretch{1.25}%  default
    \ifcase \@ptsize \relax % 10pt
```

```
      \setstretch {1.25}%
    \or % 11pt
      \setstretch {1.213}%
    \or % 12pt
      \setstretch {1.241}%
    \fi
  }%
}{%
  \renewcommand*{\onehalfspacing}{%
    \@tempdima=\dimexpr (\f@size pt)*1500/
                        (\dimexpr \f@baselineskip\relax)*\p@/1000\relax
    \expandafter\setstretch\expandafter{\strip@pt\@tempdima}%
  }%
}{%

  \scr@ifexpected{\onehalfspacing}{%
    \long\def\onehalfspacing{%
      \setstretch{1.25}%  default
% DPC 6.7b guard \@ptsize
      \ifx\@ptsize\@undefined\else
        \ifcase \@ptsize \relax % 10pt
          \setstretch {1.25}%
        \or % 11pt
          \setstretch {1.213}%
        \or % 12pt
          \setstretch {1.241}%
        \fi
      \fi
    }%
  }{%
    \renewcommand*{\onehalfspacing}{%
      \@tempdima=\dimexpr (\f@size pt)*1500/
                          (\dimexpr \f@baselineskip\relax)*\p@/1000\relax
      \expandafter\setstretch\expandafter{\strip@pt\@tempdima}%
    }%
  }{%
    \PackageWarning{scrhack}{unknown \string\onehalfspacing\space
      definition found!\MessageBreak
      Maybe you are using a unsupported setpace version}%
  }%
}
```

`\doublespacing` This selects the double line spacing. I think, a factor with three digits for the decimals should be enough.

```
\scr@ifexpected{\doublespacing}{%
  \long\def\doublespacing{%
    \setstretch{1.667}%  default
    \ifcase \@ptsize \relax % 10pt
      \setstretch {1.667}%
    \or % 11pt
      \setstretch {1.618}%
    \or % 12pt
      \setstretch {1.655}%
    \fi
  }%
}{%
  \renewcommand*{\doublespacing}{%
    \@tempdima=\dimexpr (\f@size pt)*2000/
                        (\dimexpr \f@baselineskip\relax)*\p@/1000\relax
    \expandafter\setstretch\expandafter{\strip@pt\@tempdima}%
  }%
}{%

  \scr@ifexpected{\doublespacing}{%
    \long\def\doublespacing{%
      \setstretch {1.667}%  default
% DPC 6.7b guard \@ptsize
      \ifx\@ptsize\@undefined\else
        \ifcase \@ptsize \relax % 10pt
          \setstretch {1.667}%
        \or % 11pt
          \setstretch {1.618}%
        \or % 12pt
          \setstretch {1.655}%
        \fi
      \fi
    }%
  }{%
    \renewcommand*{\doublespacing}{%
      \@tempdima=\dimexpr (\f@size pt)*2000/
                          (\dimexpr \f@baselineskip\relax)*\p@/1000\relax
      \expandafter\setstretch\expandafter{\strip@pt\@tempdima}%
    }%
  }{%
    \PackageWarning{scrhack}{unknown \string\doublespacing\space
      definition found!\MessageBreak
      Maybe you are using a unsupported setpace version}%
  }%
}
⟨/setspace & body⟩
```

2.9 The lscape hack

The package does change `\textheight` inside `landscape` to the prior value of `\textwidth`, nevertheless is does not change `\textwidth` to the prior value of `\textheight`. This is at least inconsistent. After asking him, David Carlisle answered, that he does not change `\textwidth`, because this could result in problems. Therefore it is very strange, that he does change `\textheight`, despite this also to problems, e.g., with packages showframe or scrlayer. So the patch here also keeps `\textheight` unchanged. This also has to take care for pdflscape. The patches are done using xpatch.

`lscape` (*opt.*)

```
⟨*package & option⟩
\RequirePackage{xpatch}%
\KOMA@ifkey{lscape}{@scrhack@lscape}%
\KOMAExecuteOptions{lscape=true}%
⟨/package & option⟩
⟨*package & body⟩
\AfterPackage*{lscape}{%
  \if@scrhack@lscape\scr@hack@load\@pkgextension{lscape}\else
    \KOMA@key[scrhack.sty]{lscape}{%
      \PackageWarning{scrhack}{option ‘lscape’ ignored}%
      \FamilyKeyStateProcessed
    }%
  \fi
}
⟨/package & body⟩
```

`\landscape` This command is the beginning of environment `landscape`. This has to be patched using xpatch.

Note: This is a load time option and cannot be changed after loading the package!

```
⟨*lscape & body⟩
\xpatchcmd{\landscape}{\textheight=\vsize}{%
  \if@scrhack@lscape
```

`\@outputpage` Here the changed value of `\textheight` is needed for initialization of `\@colht`. So we patch it to use the new internal `\scr@LT@textheight` instead of `\textheight`.

```
    \scrh@LT@textheight=\vsize
    \let\scrh@LT@outputpage\@outputpage
    \def\@outputpage{\scrh@LT@outputpage\global\@colht\scrh@LT@textheight}%
  \else
    \textheight=\vsize
  \fi
}{%
  \PackageInfo{scrhack}{\string\landscape\space patched to make
    \string\textheight\space change optional}%
}{%
  \PackageWarning{scrhack}{Cannot patch \string\landscape!\MessageBreak
    Maybe you are using a unsupported lscape version}%
  \@scrhack@lscapefalse
}
```

`\scrh@LT@textheight` (*ilen.*)

```
\newlength{\scrh@LT@textheight}
⟨/lscape & body⟩
```

2.10 The standard classes sections hack

The *standard classes sections hack* differs from each other hack. First of all it does not patch a specific package but the class definitions of section commands from `\part` till `\subparagraph`. Additionally it resets several internal LaTeX macros which are changed by KOMA-Script classes to their original definition. Last but not least it is not activated by default, but the user has to activate it, if he needs it. The hack has more potential to break things than to make things better. Because of this, there is not any support for using this hack and the hack shows several warning messages.

`standardsections` (*opt.*)

```
⟨*package & option⟩
\KOMA@ifkey{standardsections}{@scrhack@standardsections}
\KOMAExecuteOptions{standardsections=false}
⟨/package & option⟩
⟨*package & body⟩
\AtEndOfPackage{%
  \if@scrhack@standardsections
    \PackageInfo{scrhack}{loading standard classes sections hack}%
    \input{standardsections.hak}%
  \fi
}
⟨/package & body⟩
```

`\part`
`\chapter`
`\section`
`\subsection`
`\subsubsection`
`\paragraph`
`\subparagraph`

The hack redefines all these commands to a definition copied from the standard LaTeX classes version v1.4i. Copyright of this code is by the LaTeX3 Project. The complete, unmodified code can be found on https://ctan.org/tex-archive/macros/latex/base/classes.dtx.

```
⟨*standardsections & body⟩
\PackageWarningNoLine{scrhack}{%
  You are loosing every kind of support because of\MessageBreak
  using the dangerous standard classes sections hack!\MessageBreak
  This hack can break several classes and packages.\MessageBreak
  However, sometimes it is needed to make a\MessageBreak
  package work%
}%
\PackageInfo{scrhack}{%
  Using code from the standard LaTeX classes.\MessageBreak
  For more information about this code and the\MessageBreak
  license of this code, please see:\MessageBreak
  https://ctan.org/tex-archive/macros/latex/base/classes.dtx
}%
\@ifpackageloaded{hyperref}{%
  \PackageWarning{scrhack}{this hack should not be loaded after ‘hyperref’}%
}{}
\@ifpackageloaded{titlesec}{%
  \PackageWarning{scrhack}{this hack should not be loaded after ‘titlesec’}%
}{}
\@ifpackageloaded{sectsty}{%
  \PackageWarning{scrhack}{this hack should not be loaded after ‘sectsty’}%
}{}
\scr@ifundefinedorrelax{chapter}{%
```

It seems to be an article class, so we use the code from article.

```
  \scr@ifundefinedorrelax{c@part}{\newcounter{part}}{}%
  \scr@ifundefinedorrelax{c@section}{\newcounter{section}}{}%
```

```
\scr@ifundefinedorrelax{c@subsection}{\newcounter{subsection}[section]}{}%
\scr@ifundefinedorrelax{c@subsubsection}{%
  \newcounter{subsubsection}[subsection]}{}%
\scr@ifundefinedorrelax{c@paragraph}{%
  \newcounter{paragraph}[subsubsection]}{}%
\scr@ifundefinedorrelax{c@subparagraph}{%
  \newcounter{subparagraph}[paragraph]}{}%
\renewcommand\thepart {\@Roman\c@part}%
\renewcommand\thesection {\@arabic\c@section}%
\renewcommand\thesubsection   {\thesection.\@arabic\c@subsection}%
\renewcommand\thesubsubsection{\thesubsection.\@arabic\c@subsubsection}%
\renewcommand\theparagraph    {\thesubsubsection.\@arabic\c@paragraph}%
\renewcommand\thesubparagraph {\theparagraph.\@arabic\c@subparagraph}%
\def\part{%
   \if@noskipsec \leavevmode \fi
   \par
   \addvspace{4ex}%
   \@afterindentfalse
   \secdef\@part\@spart}%
\def\@part[#1]#2{%
    \ifnum \c@secnumdepth >\m@ne
      \refstepcounter{part}%
      \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
    \else
      \addcontentsline{toc}{part}{#1}%
    \fi
    {\parindent \z@ \raggedright
     \interlinepenalty \@M
     \normalfont
     \ifnum \c@secnumdepth >\m@ne
       \Large\bfseries \partname\nobreakspace\thepart
       \par\nobreak
     \fi
     \huge \bfseries #2%
     \markboth{}{}\par}%
    \nobreak
    \vskip 3ex
    \@afterheading}%
\def\@spart#1{%
    {\parindent \z@ \raggedright
     \interlinepenalty \@M
     \normalfont
     \huge \bfseries #1\par}%
     \nobreak
     \vskip 3ex
     \@afterheading}%
\def\section{\@startsection {section}{1}{\z@}%
                                  {-3.5ex \@plus -1ex \@minus -.2ex}%
                                  {2.3ex \@plus.2ex}%
                                  {\normalfont\Large\bfseries}}%
\def\subsection{\@startsection{subsection}{2}{\z@}%
                                    {-3.25ex\@plus -1ex \@minus -.2ex}%
```

```
                                     {1.5ex \@plus .2ex}%
                                     {\normalfont\large\bfseries}}%
  \def\subsubsection{\@startsection{subsubsection}{3}{\z@}%
                                     {-3.25ex\@plus -1ex \@minus -.2ex}%
                                     {1.5ex \@plus .2ex}%
                                     {\normalfont\normalsize\bfseries}}%
  \def\paragraph{\@startsection{paragraph}{4}{\z@}%
                                    {3.25ex \@plus1ex \@minus.2ex}%
                                    {-1em}%
                                    {\normalfont\normalsize\bfseries}}%
  \def\subparagraph{\@startsection{subparagraph}{5}{\parindent}%
                                       {3.25ex \@plus1ex \@minus .2ex}%
                                       {-1em}%
                                      {\normalfont\normalsize\bfseries}}%
}{%
```

It seems to be either a book or report class.

```
  \scr@ifundefinedorrelax{c@part}{\newcounter{part}}{}%
  \scr@ifundefinedorrelax{c@chapter}{\newcounter{chapter}}{}%
  \scr@ifundefinedorrelax{c@section}{\newcounter{section}[chapter]}{}%
  \scr@ifundefinedorrelax{c@subsection}{\newcounter{subsection}[section]}{}%
  \scr@ifundefinedorrelax{c@subsubsection}{%
    \newcounter{subsubsection}[subsection]}{}%
  \scr@ifundefinedorrelax{c@paragraph}{%
    \newcounter{paragraph}[subsubsection]}{}%
  \scr@ifundefinedorrelax{c@subparagraph}{%
    \newcounter{subparagraph}[paragraph]}{}%
  \def\@chapapp{\chaptername}%
  \scr@ifundefinedorrelax{frontmatter}{%
```

It seems not to be a book class.

```
    \def\part{%
      \if@openright
        \cleardoublepage
      \else
        \clearpage
      \fi
      \thispagestyle{plain}%
      \if@twocolumn
        \onecolumn
        \@tempswatrue
      \else
        \@tempswafalse
      \fi
      \null\vfil
      \secdef\@part\@spart}

    \def\@part[#1]#2{%
        \ifnum \c@secnumdepth >-2\relax
          \refstepcounter{part}%
          \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
        \else
          \addcontentsline{toc}{part}{#1}%
        \fi
        \markboth{}{}%
        {\centering
         \interlinepenalty \@M
         \normalfont
         \ifnum \c@secnumdepth >-2\relax
           \huge\bfseries \partname\nobreakspace\thepart
           \par
           \vskip 20\p@
         \fi
         \Huge \bfseries #2\par}%
        \@endpart}
    \def\@spart#1{%
        {\centering
         \interlinepenalty \@M
         \normalfont
         \Huge \bfseries #1\par}%
        \@endpart}
    \def\@endpart{\vfil\newpage
                  \if@twoside
                   \if@openright
                    \null
                    \thispagestyle{empty}%
                    \newpage
                   \fi
                  \fi
                  \if@tempswa
                    \twocolumn
                  \fi}
    \def\chapter{\if@openright\cleardoublepage\else\clearpage\fi
                        \thispagestyle{plain}%
                        \global\@topnum\z@
                        \@afterindentfalse
                        \secdef\@chapter\@schapter}
    \def\@chapter[#1]#2{\ifnum \c@secnumdepth >\m@ne
                             \refstepcounter{chapter}%
                             \typeout{\@chapapp\space\thechapter.}%
                             \addcontentsline{toc}{chapter}%
                                       {\protect\numberline{\thechapter}#1}%
                        \else
                          \addcontentsline{toc}{chapter}{#1}%
                        \fi
                        \chaptermark{#1}%
                        \addtocontents{lof}{\protect\addvspace{10\p@}}%
                        \addtocontents{lot}{\protect\addvspace{10\p@}}%
                        \if@twocolumn
                          \@topnewpage[\@makechapterhead{#2}]%
                        \else
                          \@makechapterhead{#2}%
                          \@afterheading
                        \fi}
    \def\@makechapterhead#1{%
```

```
    \vspace*{50\p@}%
    {\parindent \z@ \raggedright \normalfont
      \ifnum \c@secnumdepth >\m@ne
          \huge\bfseries \@chapapp\space \thechapter
          \par\nobreak
          \vskip 20\p@
      \fi
      \interlinepenalty\@M
      \Huge \bfseries #1\par\nobreak
      \vskip 40\p@
    }}
  \def\@schapter#1{\if@twocolumn
                     \@topnewpage[\@makeschapterhead{#1}]%
                   \else
                     \@makeschapterhead{#1}%
                     \@afterheading
                   \fi}
  \def\@makeschapterhead#1{%
    \vspace*{50\p@}%
    {\parindent \z@ \raggedright
      \normalfont
      \interlinepenalty\@M
      \Huge \bfseries  #1\par\nobreak
      \vskip 40\p@
    }}
  \def\section{\@startsection {section}{1}{\z@}%
                                     {-3.5ex \@plus -1ex \@minus -.2ex}%
                                     {2.3ex \@plus.2ex}%
                                     {\normalfont\Large\bfseries}}%
  \def\subsection{\@startsection{subsection}{2}{\z@}%
                                       {-3.25ex\@plus -1ex \@minus -.2ex}%
                                       {1.5ex \@plus .2ex}%
                                       {\normalfont\large\bfseries}}%
  \def\subsubsection{\@startsection{subsubsection}{3}{\z@}%
                                       {-3.25ex\@plus -1ex \@minus -.2ex}%
                                       {1.5ex \@plus .2ex}%
                                       {\normalfont\normalsize\bfseries}}%
  \def\paragraph{\@startsection{paragraph}{4}{\z@}%
                                      {3.25ex \@plus1ex \@minus.2ex}%
                                      {-1em}%
                                      {\normalfont\normalsize\bfseries}}%
  \def\subparagraph{\@startsection{subparagraph}{5}{\parindent}%
                                         {3.25ex \@plus1ex \@minus .2ex}%
                                         {-1em}%
                                        {\normalfont\normalsize\bfseries}}%
}{%
```

It seems to be a book class.

```
  \scr@ifundefinedorrelax{if@mainmatter}{%
    \expandafter\newif\csname if@mainmatter\endcsname
  }{}%
  \def\frontmatter{%
```

```
    \cleardoublepage
  \@mainmatterfalse
  \pagenumbering{roman}}%
\def\mainmatter{%
    \cleardoublepage
  \@mainmattertrue
  \pagenumbering{arabic}}%
\def\backmatter{%
  \if@openright
    \cleardoublepage
  \else
    \clearpage
  \fi
  \@mainmatterfalse}%
\def\part{%
  \if@openright
    \cleardoublepage
  \else
    \clearpage
  \fi
  \thispagestyle{plain}%
  \if@twocolumn
    \onecolumn
    \@tempswatrue
  \else
    \@tempswafalse
  \fi
  \null\vfil
  \secdef\@part\@spart}%

\def\@part[#1]#2{%
    \ifnum \c@secnumdepth >-2\relax
      \refstepcounter{part}%
      \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
    \else
      \addcontentsline{toc}{part}{#1}%
    \fi
    \markboth{}{}%
    {\centering
     \interlinepenalty \@M
     \normalfont
     \ifnum \c@secnumdepth >-2\relax
       \huge\bfseries \partname\nobreakspace\thepart
       \par
       \vskip 20\p@
     \fi
     \Huge \bfseries #2\par}%
    \@endpart}%
\def\@spart#1{%
    {\centering
     \interlinepenalty \@M
     \normalfont
     \Huge \bfseries #1\par}%
    \@endpart}%
\def\@endpart{\vfil\newpage
              \if@twoside
               \if@openright
                \null
                \thispagestyle{empty}%
                \newpage
               \fi
              \fi
              \if@tempswa
                \twocolumn
              \fi}%
\def\chapter{\if@openright\cleardoublepage\else\clearpage\fi
                    \thispagestyle{plain}%
                    \global\@topnum\z@
                    \@afterindentfalse
                    \secdef\@chapter\@schapter}%
\def\@chapter[#1]#2{\ifnum \c@secnumdepth >\m@ne
                       \if@mainmatter
                         \refstepcounter{chapter}%
                         \typeout{\@chapapp\space\thechapter.}%
                         \addcontentsline{toc}{chapter}%
                                   {\protect\numberline{\thechapter}#1}%
                       \else
                         \addcontentsline{toc}{chapter}{#1}%
                       \fi
                    \else
                      \addcontentsline{toc}{chapter}{#1}%
                    \fi
                    \chaptermark{#1}%
                    \addtocontents{lof}{\protect\addvspace{10\p@}}%
                    \addtocontents{lot}{\protect\addvspace{10\p@}}%
                    \if@twocolumn
                      \@topnewpage[\@makechapterhead{#2}]%
                    \else
                      \@makechapterhead{#2}%
                      \@afterheading
                    \fi}%
\def\@makechapterhead#1{%
  \vspace*{50\p@}%
  {\parindent \z@ \raggedright \normalfont
    \ifnum \c@secnumdepth >\m@ne
      \if@mainmatter
        \huge\bfseries \@chapapp\space \thechapter
        \par\nobreak
        \vskip 20\p@
      \fi
    \fi
    \interlinepenalty\@M
    \Huge \bfseries #1\par\nobreak
    \vskip 40\p@
```

```
      }}%
    \def\@schapter#1{\if@twocolumn
                       \@topnewpage[\@makeschapterhead{#1}]%
                     \else
                       \@makeschapterhead{#1}%
                       \@afterheading
                     \fi}%
    \def\@makeschapterhead#1{%
      \vspace*{50\p@}%
      {\parindent \z@ \raggedright
        \normalfont
        \interlinepenalty\@M
        \Huge \bfseries  #1\par\nobreak
        \vskip 40\p@
      }}%
    \def\section{\@startsection {section}{1}{\z@}%
                                       {-3.5ex \@plus -1ex \@minus -.2ex}%
                                       {2.3ex \@plus.2ex}%
                                       {\normalfont\Large\bfseries}}%
    \def\subsection{\@startsection{subsection}{2}{\z@}%
                                         {-3.25ex\@plus -1ex \@minus -.2ex}%
                                         {1.5ex \@plus .2ex}%
                                         {\normalfont\large\bfseries}}%
    \def\subsubsection{\@startsection{subsubsection}{3}{\z@}%
                                         {-3.25ex\@plus -1ex \@minus -.2ex}%
                                         {1.5ex \@plus .2ex}%
                                         {\normalfont\normalsize\bfseries}}%
    \def\paragraph{\@startsection{paragraph}{4}{\z@}%
                                        {3.25ex \@plus1ex \@minus.2ex}%
                                        {-1em}%
                                        {\normalfont\normalsize\bfseries}}%
    \def\subparagraph{\@startsection{subparagraph}{5}{\parindent}%
                                           {3.25ex \@plus1ex \@minus .2ex}%
                                           {-1em}%
                                          {\normalfont\normalsize\bfseries}}%
  }%
}% End of the derived code.
```

`\@startsection` KOMA-Script classes redefine this internal LaTeX kernel macro. In this case `\scr@saved@startsection` is defined, differs from `\@startsection` and has the definition found before redefining the original macro. We do nothing else but resetting `\@startsection` to this saved definition.

```
\scr@ifundefinedorrelax{scr@saved@startsection}{}{%
  \ifx\@startsection\scr@saved@startsection\else
    \PackageWarningNoLine{scrhack}{%
      Resetting \string\@startsection,\MessageBreak
      because of loading standard classes sections hack%
    }%
    \let\@startsection\scr@saved@startsection
  \fi
}
```

`\@sect` `\@ssect` `\@xsect` All these should be resetted to their original definitions stored in `\scr@latex@...`.

```
\scr@ifundefinedorrelax{scr@latex@sect}{}{%
  \ifx\@sect\scr@latex@sect\else
    \PackageWarningNoLine{scrhack}{%
      Resetting \string\@sect,\MessageBreak
      because of loading standard classes sections hack%
    }%
    \let\@sect\scr@latex@sect
  \fi
}
\scr@ifundefinedorrelax{scr@latex@ssect}{}{%
  \ifx\@ssect\scr@latex@ssect\else
    \PackageWarningNoLine{scrhack}{%
      Resetting \string\@ssect,\MessageBreak
      because of loading standard classes sections hack%
    }%
    \let\@ssect\scr@latex@ssect
  \fi
}
\scr@ifundefinedorrelax{scr@latex@xsect}{}{%
  \ifx\@xsect\scr@latex@xsect\else
    \PackageWarningNoLine{scrhack}{%
      Resetting \string\@xsect,\MessageBreak
      because of loading standard classes sections hack%
    }%
    \let\@xsect\scr@latex@xsect
  \fi
}
```

`\DeclareSectionCommand`
`\RedeclareSectionCommand`
`\DeclareNewSectionCommand`
`\ProvideSectionCommand`
`\DeclareSectionCommands`
`\RedeclareSectionCommands`
`\DeclareNewSectionCommands`
`\ProvideSectionCommands`

These KOMA-Script commands do not make sense any longer after forced redefinition of `\part` … `\subparagraph`. They even would not work correctly after resetting, e.g., `\@startsection`. So best reaction is to *undefine* them.

```
\scr@ifundefinedorrelax{DeclareSectionCommand}{}{%
  \PackageWarningNoLine{scrhack}{%
    \string\DeclareSectionCommand\space
    and all derivates deactivated,\MessageBreak
    because of loading standard classes sections hack%
  }%
  \let\DeclareSectionCommand\undefined
  \let\DeclareSectionCommands\undefined
  \let\RedeclareSectionCommand\undefined
  \let\RedeclareSectionCommands\undefined
  \let\DeclareNewSectionCommand\undefined
  \let\DeclareNewSectionCommands\undefined
  \let\ProvideSectionCommand\undefined
  \let\ProvideSectionCommands\undefined
}
```

`r@chapter@before@hyperref@patch`
`cr@chapter@after@hyperref@patch`

Now, definitions without respecting hyperref are used, so hyperref should again be able to patch them. So the hack should deactivate the patch deactivation of the KOMA-Script classes.

```
\let\scr@chapter@before@hyperref@patch\relax
```

```
\let\scr@chapter@after@hyperref@patch\relax
```

`\scr@osectarg` The KOMA-Script classes use this macro to store how to use the optional argument of section commands. But with the standard class definition the only possibility is to use it as running head and write it into the table of contents without any extended handling. So the only value, that makes sense is 0.

```
\scr@ifundefinedorrelax{scr@osectarg}{}{%
  \def\reserved@a{0}%
  \ifx\scr@osectarg\reserved@a\else
    \let\scr@osectarg\reserved@a
    \PackageWarningNoLine{scrhack}{%
      Extended option handling for section commands deactivated,\MessageBreak
      because of loading standard classes sections hack%
    }%
  \fi
}%
```

`\scr@activate@xsection` Further changing of the value has to be deactivated.

```
\scr@ifundefinedorrelax{scr@activate@xsection}{}{%
  \renewcommand*{\scr@activate@xsection}[1]{%
    \edef\scr@osectarg{#1}\def\reserved@a{0}%
    \ifx\scr@osectarg\reserved@a
    \else
      \let\scr@osectarg\reserved@a
      \PackageWarning{scrhack}{%
        Because of loading the standard classes\MessageBreak
        sections hack, the extended option handling\MessageBreak
        of section commands failed%
      }%
    \fi
  }%
}
```

`\partformat`
`\chapterformat` Most `\...format` macros of KOMA-Script are still used, because of the redefinition of `\@seccntformat`. But `\part` and `\chapter` are defined without using them.

```
\scr@ifundefinedorrelax{partformat}{}{%
  \PackageWarningNoLine{scrhack}{%
    Usage of \string\partformat\space
    deactivated,\MessageBreak
    because of loading the standard classes sections hack%
  }%
}
\scr@ifundefinedorrelax{chapterformat}{}{%
  \PackageWarningNoLine{scrhack}{%
    Usage of \string\chapterformat\space
    deactivated,\MessageBreak
    because of loading the standard classes sections hack%
  }%
}
```

`\sectionformat`
`\subsectionformat`
`\subsubsectionformat`
`\paragraphformat`
`\subparagraphformat`

All these commands have to be resetted to not use the automatic dot and to add a `\quad` instead of an `\enskip`.

```
\scr@ifundefinedorrelax{sectionformat}{}{%
  \PackageWarningNoLine{scrhack}{%
    \string\sectionformat\space set to
    `\string\thesection\string\quad',\MessageBreak
    because of loading the standard classes sections hack%
  }%
  \renewcommand*{\sectionformat}{\thesection\quad}%
}
\scr@ifundefinedorrelax{subsectionformat}{}{%
  \PackageWarningNoLine{scrhack}{%
    \string\subsectionformat\space set to
    `\string\thesubsection\string\quad'\MessageBreak
    because of loading the standard classes sections hack%
  }%
  \renewcommand*{\subsectionformat}{\thesubsection\quad}%
}
\scr@ifundefinedorrelax{subsubsectionformat}{}{%
  \PackageWarningNoLine{scrhack}{%
    \string\subsubsectionformat\space set to
    `\string\thesubsubsection\string\quad'\MessageBreak
    because of loading the standard classes sections hack%
  }%
  \renewcommand*{\subsubsectionformat}{\thesubsubsection\quad}%
}
\scr@ifundefinedorrelax{paragraphformat}{}{%
  \PackageWarningNoLine{scrhack}{%
    \string\paragraphformat\space set to
    `\string\theparagraph\string\quad'\MessageBreak
    because of loading the standard classes sections hack%
  }%
  \renewcommand*{\paragraphformat}{\theparagraph\quad}%
}
\scr@ifundefinedorrelax{subparagraphformat}{}{%
  \PackageWarningNoLine{scrhack}{%
    \string\subparagraphformat\space set to
    `\string\thesubparagraph\string\quad'\MessageBreak
    because of loading the standard classes sections hack%
  }%
  \renewcommand*{\subparagraphformat}{\thesubparagraph\quad}%
}
```

`\partlineswithprefixformat`
`\chapterlineswithprefixformat`
`\chapterlinesformat`
`\sectionlinesformat`
`\sectioncatchphraseformat`

These KOMA-Script commands are not used any longer.

```
\scr@ifundefinedorrelax{partlineswithprefixformat}{}{%
  \PackageWarningNoLine{scrhack}{%
    Usage of \string\partlineswithprefixformat\space deactivated,\MessageBreak
    because of loading the standard classes sections hack%
  }%
}
\scr@ifundefinedorrelax{chapterlineswithprefixformat}{}{%
```

```
  \PackageWarningNoLine{scrhack}{%
    Usage of \string\chapterlineswithprefixformat\space and\MessageBreak
    \string\chapterlinesformat\space deactivated,\MessageBreak
    because of using standard classes sections hack%
  }%
}
\scr@ifundefinedorrelax{sectionlinesformat}{}{%
  \PackageWarningNoLine{scrhack}{%
    Usage of \string\sectionlinesformat\space and\MessageBreak
    \string\sectioncatchphraseformat\space deactivated,\MessageBreak
    because of using standard classes sections hack%
  }%
}
```

`headings` (*opt.*) The KOMA-Script option does not make sense any longer. So it is deactivated and usage will throw an error.

```
\scr@ifundefinedorrelax{KOMAClassName}{}{%
  \PackageWarningNoLine{scrhack}{%
    KOMA-Script option ‘headings’ removed,\MessageBreak
    because of loading standard classes sections hack%
  }%
  \RelaxFamilyKey[.\KOMAClassFileName]{KOMA}{headings}%
  \KOMA@kav@removekey{\KOMAClassFileName}{headings}%
}
```

`\scr@class@titlesec@warning` The warning about loading titlesec has to be changes, because we do not expect errors any longer.

```
\def\scr@class@titlesec@warning{%
  \PackageWarning{scrhack}{%
    Usage of ‘titlesec’ should work, because of\MessageBreak
    loading the standard classes sections hack.\MessageBreak
    However, you should know, there is not any\MessageBreak
    support for this combination by the\MessageBreak
    KOMA-Script author%
  }%
}
⟨/standardsections & body⟩
```

2.11 Executing options

Last but not least the options have to be processed. In the generated package this will be done before the commands of subsection 2.2.

```
⟨*package & option⟩
\KOMAProcessOptions\relax
⟨/package & option⟩
```

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.

Change History

- 3.23 – 2017/03/31
  - \if@intoc: support of \if@intoc 16
- v3.03 – 2009/03/12
  - General: first version ever 1
- v3.04b – 2009/08/05
  - General: order of commands/macros and options changed 1
- v3.04b – 2009/11/09
  - hyperref: loading hyperref hack earlier . 3
- v3.12 – 2013/03/05
  - float: using \FamilyKeyStateProcessed 9
  - floatrow: using \FamilyKeyStateProcessed 6
  - hyperref: using \FamilyKeyStateProcessed 3
  - listings: using \FamilyKeyStateProcessed 11
- v3.17 – 2015/03/09
  - float: using \KOMAExecuteOptions . . . 9
  - floatrow: using \KOMAExecuteOptions . 6
  - hyperref: auto deactivation for newer versions of hyperref 3
    - defaults by \KOMAExecuteOptions . . . 3
  - listings: using \KOMAExecuteOptions 11
- v3.17 – 2015/04/16
  - \doublespacing: new 18
  - \onehalfspacing: added 17
  - setspace: added 17
- v3.18 – 2015/05/22
  - hyperref: newer versions of KOMA-Script classes also deactivate the the hack 3
- v3.18a – 2015/07/07
  - General: lscape hack added 19
  - \landscape: added 20
  - lscape: added 19
- v3.18a – 2015/08/11
  - \@outputpage: added 20
  - \scrh@LT@textheight: new 20
- v3.20 – 2016/05/07
  - \landscape: loading xpatch in option part already 20
  - lscape: loading xpatch in option part already 19
- v3.23 – 2017/03/31
  - nomencl: added 14
  - \scr@hack@nomencl@head: heading of thenomenclature by tocbasic 16
  - \thenomenclature: added 14
- v3.25 – 2018/03/16
  - \listof: using \DeclareTOCStyleEntry 7
  - \newfloat: using \DeclareTOCStyleEntry 9
- v3.25 – 2019/01/23
  - \thenomenclature: emergency patch added 15
- v3.26b – 2019/01/23
  - \thenomenclature: new nomencl 5.0 . . 15
    - prepared for hoply comming nomencl 15
- v3.28 – 2019/11/19
  - \newfloat: \ifattoclist replaced by \Ifattoclist 9
- v3.30 – 2020/02/25
  - \thenomenclature: spurious space in warning message removed 15
- v3.30 – 2020/04/01
  - \lst@makecaption: setting of \@captype added 13

- v3.31 – 2020/07/08
  - `\scr@hack@nomencl@head`: internal `\list@fname` eliminated 16
- v3.34 – 2021/06/03
  - `floatrow`: deactivation of the float hack 6
- v3.36 – 2022/02/08
  - General: require package scrlogo instead of defining `\KOMAScript` 1
    - switch over from scrdoc to koma-script-source-doc 1
    - whole implementation documentation in English 1
- v3.39 – 2022/11/11
  - `float`: initial dot in member argument of `\KOMA@key` removed 9
  - `floatrow`: initial dot in member argument of `\KOMA@key` removed 6
  - `headings`: initial dot in member argument of option storage commands removed 32
  - `hyperref`: initial dot in member argument of `\KOMA@key` and `\KOMAExecuteOptions` removed 3
  - `listings`: initial dot in member argument of `\KOMA@key` removed 11
  - `lscape`: initial dot in member argument of `\KOMA@key` removed 19
  - `nomencl`: initial dot in member argument of `\KOMA@key` removed 14
  - `setspace`: initial dot in member argument of `\KOMA@key` removed 17
- v3.39 – 2023/03/15
  - `\doublespacing`: support for setspace 6.7b added 19
  - `\onehalfspacing`: support for setspace 6.7b added 18
- v3.40 – 2023/04/17
  - General: guide names changed 1
- v3.41 – 2023/06/28
  - `\scr@do@hack@listings`: patching `\lst@MakeCaption` to use `\ext@lstlisting` 12
- v3.42 – 2024/04/17
  - `listings`: automatic deactivation for new listings 11