

Legacy Version of Package scrhack

Markus Kohm

2023-07-15

Contents

1 User Manual	1
2 Implementation of the Legacy Version of scrhack	1
2.1 Options	1
2.2 Used Macros and Commands	2
2.3 The hyperref hack	3
2.4 The floatrow hack	6
2.5 The float hack	9
2.6 The listings hack	11
2.7 The nomenc1 hack	14
2.8 The setspace hack	16
2.9 The lscape hack	19
2.10 The standard classes sections hack	20
2.11 Executing options	32
Index	33
Change History	34

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:

```
1 (*package & option)
2 \RequirePackage{scrkbase}[2022-11-11]
3 \@ifpackagelater{scrkbase}{2023-07-14}{%
4   \PackageInfo{scrhack}{KOMA-Script >= 3.42 detected.\MessageBreak
5     In this case you should not use this legacy version of\MessageBreak
```

```

6   scrhack, but should be able to use the current version\MessageBreak
7   or even replace loading scrhack by loading really\MessageBreak
8   needed third-party patches, e.g., floatbyctobasic.\MessageBreak
9   But you are the boss ..}%
10 }{%
11  \@ifpackagelater{scrkbase}{2022-11-11}{%
12   \PackageWarningNoLine{scrhack}{KOMA-Script < 3.42 detected.\MessageBreak
13    In this case the original scrhack of that version\MessageBreak
14    should also be installed and used instead of this\MessageBreak
15    legacy package}%
16 }{%
17  \PackageError{scrhack}{incompatible KOMA-Script < v3.39 detected}{%
18   This error is fatal!\MessageBreak
19   Loading of scrhack will be aborted!}%
20  \expandafter\let\csname ver@scrhack.sty\endcsname\undefined
21  \endinput
22 }%
23 }
24 </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

```
25 <*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!

```

26 \newcommand{\scr@ifexpected}[2]{%
27  \begingroup
28   \let\@tempa#1
29   #2
30   \ifx\@tempa#1
31    \aftergroup\@firstoftwo
32   \else
33    \aftergroup\@secondoftwo
34   \fi
35  \endgroup
36 }

```

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

```
\scr@hack@load{<extension>}{<basename>}{<hack>} {<true code>}{<false code>}
```

The first two arguments $\langle \rangle$ and $\langle \textit{extension} \rangle$ and $\langle \textit{basename} \rangle$ are same to the internal L^AT_EX kernel command `\@ifl@aded`. With L^AT_EX 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 L^AT_EX kernels, we test the definition of `\ver@ $\langle \textit{basename} \rangle$. $\langle \textit{extension} \rangle$` . If it is defined, we also test the definition of `\ver@ $\langle \textit{basename} \rangle$. $\langle \textit{hack} \rangle$. $\langle \textit{hack} \rangle$` . If this is not defined, `\langle \textit{hack} \rangle`. $\langle \textit{hack} \rangle$ is loaded and $\langle \textit{true code} \rangle$ is executed. Otherwise $\langle \textit{false code} \rangle$ 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.

```

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

```

Load a hack only once.

```

41   \expandafter\ifx\csname ver@#2.hak\endcsname\relax
42     \expandafter\expandafter\expandafter\@firstoftwo
43   \else
44     \expandafter\expandafter\expandafter\@secondoftwo
45   \fi
46 \fi
47 {%
48   \PackageInfo{scrhack}{loading #2 hack}%
49   \edef\reserved@a{%
50     \noexpand\makeatletter\noexpand\input{#2.hak}%
51     \noexpand\catcode'\noexpand\@the\catcode'\@relax
52   }\reserved@a
53 }{%
54   \PackageInfo{scrhack}{ignoring #2 hack}%
55 }%
56 }

57 </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.*)

```

58 <*package & option>
59 \KOMA@ifkey{hyperref}{\scrhack@hyperref}%
60 \KOMAEecuteOptions{hyperref=true}%
61 </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.

```

62 (*package & body)
63 \BeforePackage{hyperref}{%
64   \scr@ifundefinedorrelax{hy@insteadofrefstepcounter}{}{%
65     \PackageInfo{scrhack}{hyperref hack deactivated because of\MessageBreak
66       detection of KOMA-Script class, that doesn't\MessageBreak
67       need that hack,}%
68     \KOMAEecuteOptions[scrhack.sty]{hyperref=false}%
69   }%
70 }
71 \AfterPackage*{hyperref}{%
72   \if@scrhack@hyperref
73     \@ifpackagelater{hyperref}{2009/11/24}{%
74       \PackageInfo{scrhack}{hyperref hack deactivated because of\MessageBreak
75         detection of hyperref version, that doesn't\MessageBreak
76         need that hack,}%
77       \KOMAEecuteOptions[scrhack.sty]{hyperref=false}%
78     }{%
79       \AtEndOfPackage{%
80         \KOMA@key[scrhack.sty]{hyperref}[true]{%
81           \PackageWarning{scrhack}{option 'hyperref=#1' ignored}%
82           \FamilyKeyStateProcessed
83         }%
84         \if@scrhack@hyperref\scr@hack@load\@pkgextension{hyperref}\fi
85       }%
86     }%
87   \fi
88 }
89 \end{package & body}

```

`\@schapter` We do not hack `hyperref` itself, but the driver files. After loading `hyperref` the driver file is `\@spart` also loaded. Moreover, all driver files that re-define these commands have to be patched `\@ssect` 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.

```

90 (*hyperref & body)
91 \scr@ifexpected\@schapter{%
92   \def\@schapter#1{%
93     \H@old@schapter{#1}%
94     \begingroup
95       \let\@mkboth\@gobbletwo
96       \Hy@GlobalStepCount\Hy@linkcounter
97       \xdef\@currentHref{\Hy@chapapp*.\the\Hy@linkcounter}%
98       \Hy@raisedlink{%
99         \hyper@anchorstart{\@currentHref}\hyper@anchorend
100      }%
101     \endgroup
102   }%
103 }{%
104   \PackageInfo{scrhack}{redefining \string\@schapter}%

```

```

105 \def\@schapter#1{%
106 \begingroup
107 \let\@mkboth\@gobbletwo
108 \Hy@GlobalStepCount\Hy@linkcounter
109 \xdef\@currentHref{\Hy@chapapp*.\the\Hy@linkcounter}%
110 \Hy@raisedlink{%
111 \hyper@anchorstart{\@currentHref}\hyper@anchorend
112 }%
113 \endgroup
114 \H@old@schapter{#1}%
115 }%
116 }{%
117 \scr@ifexpected\@schapter{%
118 \def\@schapter#1{%
119 \begingroup
120 \let\@mkboth\@gobbletwo
121 \Hy@GlobalStepCount\Hy@linkcounter
122 \xdef\@currentHref{\Hy@chapapp*.\the\Hy@linkcounter}%
123 \Hy@raisedlink{%
124 \hyper@anchorstart{\@currentHref}\hyper@anchorend
125 }%
126 \endgroup
127 \H@old@schapter{#1}%
128 }%
129 }{%
130 \PackageWarningNoLine{scrhack}{unknown \string\@schapter\space
131 definition found!\MessageBreak
132 Maybe you are using a unsupported hyperref version}%
133 }%
134 }
135 \scr@ifexpected\@spart{%
136 \def\@spart#1{%
137 \H@old@spart{#1}%
138 \Hy@GlobalStepCount\Hy@linkcounter
139 \xdef\@currentHref{part*.\the\Hy@linkcounter}%
140 \Hy@raisedlink{%
141 \hyper@anchorstart{\@currentHref}\hyper@anchorend
142 }%
143 }%
144 }{%
145 \PackageInfo{scrhack}{redefining \string\@spart}%
146 \def\@spart#1{%
147 \Hy@GlobalStepCount\Hy@linkcounter
148 \xdef\@currentHref{part*.\the\Hy@linkcounter}%
149 \Hy@raisedlink{%
150 \hyper@anchorstart{\@currentHref}\hyper@anchorend
151 }%
152 \H@old@spart{#1}%
153 }%
154 }{%
155 \scr@ifexpected\@spart{%
156 \def\@spart#1{%

```

```

157     \Hy@GlobalStepCount\Hy@linkcounter
158     \xdef\@currentHref{part*.\the\Hy@linkcounter}%
159     \Hy@raisedlink{%
160       \hyper@anchorstart{\@currentHref}\hyper@anchorend
161     }%
162     \H@old@spart{#1}%
163   }%
164 }-{}-%
165   \PackageWarningNoLine{scrhack}{unknown \string\@spart\space
166     definition found!\MessageBreak
167     Maybe you are using a unsupported hyperref version}%
168 }%
169 }
170 \scr@ifexpected\@ssect{%
171   \def\@ssect#1#2#3#4#5{%
172     \H@old@ssect{#1}{#2}{#3}{#4}{#5}%
173     \phantomsection
174   }%
175 }-{}-%
176   \PackageInfo{scrhack}{redefining \string\@ssect}%
177   \def\@ssect#1#2#3#4#5{%
178     \H@old@ssect{#1}{#2}{#3}{#4}{\phantomsection\ignorespaces#5}%
179   }%
180 }-{}-%
181   \scr@ifexpected\@ssect{%
182     \def\@ssect#1#2#3#4#5{%
183       \H@old@ssect{#1}{#2}{#3}{#4}{\phantomsection\ignorespaces#5}%
184     }%
185   }-{}-%
186   \PackageWarningNoLine{scrhack}{unknown \string\@ssect\space
187     definition found!\MessageBreak
188     Maybe you are using a unsupported hyperref version}%
189 }%
190 }
191 </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.*)

```

192 <*package & option>
193 \KOMA@ifkey{floatrow}{@scrhack@floatrow}
194 \KOMAExecuteOptions{floatrow=true}
195 </package & option>

```

```

196 <*package & body>
197 \AfterPackage*{floatrow}{%
198   \KOMAOptions{float=false}%
199   \PackageInfo{scrhack}{option 'float' deactivated}%
200   \KOMA@key[scrhack.sty]{floatrow}{%
201     \PackageWarning{scrhack}{option 'floatrow' ignored}%
202     \FamilyKeyStateProcessed
203   }%
204   \if@scrhack@floatrow\scrhack@load\@pkgextension{floatrow}\fi
205 }
206 </package & body>

```

`\DeclareNewFloatType` 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 and will be reset to the original definition.

```

207 <*floatrow & body>
208 \scr@ifexpected{\DeclareNewFloatType}{%
209   \long\def\DeclareNewFloatType#1#2{\def\FB@captype{#1}%
210     \expandafter\edef\csname ftype@#1\endcsname{\the\c@float@type}%
211     \addtocounter{float@type}{\value{float@type}}%
212     \@namedef{#1name}{#1}\newcounter{#1}%
213     \expandafter\edef\csname fnum@#1\endcsname
214     {\expandafter\noexpand\csname #1name\endcsname\nobreakspace
215     \expandafter\noexpand\csname the#1\endcsname}%
216     \@namedef{the#1}{\arabic{#1}}\flnew@ext{lo#1}\@namedef{fps@#1}{tbp}%
217     \@namedef{l@#1}{\@dottedtocline{1}{1.5em}{2.3em}}%
218     \caption@setkeys[floatrow]{newfloat}{#2}\let\FR@tmp=\relax
219     \xdef\@tempa{\noexpand\flrow@types{\the\flrow@types \FR@tmp{#1}}}%
220     \@tempa}%
221 }{%
222   \scr@ifexpected{\listof}{%
223     \def\listof#1#2{%
224       \ifundefined{ext@#1}{\flrow@error{Unknown float style '#1'}}{%
225         \expandafter\providecommand\csname l@#1\endcsname
226         {\@dottedtocline{1}{1.5em}{2.3em}}%
227         \float@listhead{#2}%
228         \begingroup\setlength{\parskip}{\z@}%
229         \@starttoc{\@nameuse{ext@#1}}%
230         \endgroup}}%
231   }{%
232     \RequirePackage{tocbasic}%
233     \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!

```

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

```

```

235 \expandafter\edef\csname ftype@#1\endcsname{\the\c@float@type}%
236 \addtocounter{float@type}{\value{float@type}}%
237 \@namedef{#1name}{#1}\newcounter{#1}%
238 \expandafter\edef\csname fnum@#1\endcsname
239 {\expandafter\noexpand\csname #1name\endcsname\nobreakspace
240 \expandafter\noexpand\csname the#1\endcsname}%
241 \@namedef{the#1}{\arabic{#1}}\flnew@ext{lo#1}\@namedef{fps@#1}{tbp}%
242 \DeclareTOCStyleEntry[level=1,numwidth=2.3em,indent=1.5em]{default}{#1}%
243 \caption@setkeys[floatrow]{newfloat}{#2}\let\FR@tmp=\relax
244 \xdef\@tempa{\noexpand\flrow@types{\the\flrow@types\FR@tmp{#1}}}%
245 \@tempa
246 \xdef\@tempa{\noexpand\addtotoclist[float]{\@nameuse{ext@FB@capttype}}%
247 \noexpand\setuptoc{\@nameuse{ext@FB@capttype}}{chapteratlist}%
248 }%
249 \@tempa
250 }%
251 \PackageInfo{scrhack}{redefining \string\listof}%
252 \renewcommand*\listof[2]{%
253 \ifundefined{ext@#1}{\flrow@error{Unknown float style '#1'}}{%
254 \ifundefined{l@#1}{\expandafter\let\csname l@#1\endcsname=l@figure
255 \ifundefined{l@#1}{%
256 \DeclareTOCStyleEntry[level=1,numwidth=2.3em,indent=1.5em]
257 {default}{#1}%
258 }{%
259 }{%
260 \listoftoc[#{2}]{\csname ext@#1\endcsname}%
261 }%
262 }%
263 \scr@ifexpected{\float@addtolists}{%
264 \long\def\float@addtolists#1{%
265 \def\float@do##1{\addtocontents{##1}{#1}} \the\float@exts}%
266 }{%
267 \PackageInfo{scrhack}{undefining \string\float@addtolists}%
268 \let\float@addtolists\relax
269 }{%
270 \PackageWarningNoLine{scrhack}{unkown \string\float@addtolists\space
271 definition found!\MessageBreak
272 Maybe you are using a unsupported floatrow version}%
273 }%
274 }{%
275 \PackageWarningNoLine{scrhack}{unknown \string\listof\space
276 definition found!\MessageBreak
277 Maybe you are using a unsupported floatrow version}%
278 }%
279 }{%
280 \PackageWarningNoLine{scrhack}{unknown \string\DeclareNewFloatType\space
281 definition found!\MessageBreak
282 Maybe you are using a unsupported floatrow version}%
283 }
284 </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.*)

```
285 (*package & option)
286 \KOMA@ifkey{float}{@scrhack@float}%
287 \KOMAEecuteOptions{float=true}%
288 (/package & option)
289 (*package & body)
290 \AfterPackage*{float}{-%
291   \KOMA@key[scrhack.sty]{float}{-%
292     \PackageWarning{scrhack}{option 'float' ignored}%
293     \FamilyKeyStateProcessed
294   }%
295   \if@scrhack@float\scr@hack@load\@pkgextension{float}\fi
296 }
297 (/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@addtolist` The deprecated macro `\float@addtolist` is not needed any longer and will be reset to the original definition.

```
298 (*float & body)
299 \scr@ifexpected{\newfloat}{-%
300   \long\def\newfloat#1#2#3{\@namedef{ext@#1}{#3}
301     \let\float@do=\relax
302     \xdef\@tempa{\noexpand\float@exts{\the\float@exts \float@do{#3}}}%
303     \@tempa
304     \floatplacement{#1}{#2}%
305     \@ifundefined{fname@#1}{\floatname{#1}{#1}}{}
306     \expandafter\edef\csname ftype@#1\endcsname{\value{float@type}}%
307     \addtocounter{float@type}{\value{float@type}}
308     \restylefloat{#1}%
309     \expandafter\edef\csname fnum@#1\endcsname%
310     {\expandafter\noexpand\csname fname@#1\endcsname}
311     \expandafter\noexpand\csname the#1\endcsname}
312   \@ifnextchar[%
313     {\float@newx{#1}}%
314     {\@ifundefined{c@#1}{\newcounter{#1}\@namedef{the#1}{\arabic{#1}}}%
315     {}}}%
```

```

316 }-%
317 \scr@ifexpected{\listof}{-%
318   \def\listof#1#2{%
319     \@ifundefined{ext@#1}{\float@error{#1}}{%
320       \@namedef{l@#1}{\@dottedtocline{1}{1.5em}{2.3em}}%
321       \float@listhead{#2}%
322       \begingroup\setlength{\parskip}{\z@}%
323       \@starttoc{\@nameuse{ext@#1}}%
324       \endgroup}}%
325 }-%
326 \RequirePackage{tocbasic}%
327 \PackageInfo{scrhack}{redefining \string\newfloat}%
328 \renewcommand\newfloat[3]{%
329   \Ifattoclist{#3}{-%
330     \PackageError{scrhack}{extension ‘#3’ already in use}{-%
331       Each extension may be used only once.\MessageBreak
332       You, the class, or another package already uses extension
333       ‘#3’.\MessageBreak
334       \string\newfloat\space command will be ignored!}%
335   }-%
336   \addtotoclist[float]{#3}%
337   \setuptoc{#3}{chapteratlist}%
338   \@namedef{ext@#1}{#3}%
339   \let\float@do=\relax
340   \xdef\@tempa{\noexpand\float@exts{\the\float@exts \float@do{#3}}}%
341   \@tempa
342   \floatplacement{#1}{#2}%
343   \@ifundefined{fname@#1}{\floatname{#1}{#1}}{%
344     \expandafter\edef\csname ftype@#1\endcsname{\value{float@type}}%
345     \addtocounter{float@type}{\value{float@type}}
346     \restylefloat{#1}%
347     \expandafter\edef\csname fnum@#1\endcsname%
348     {\expandafter\noexpand\csname fname@#1\endcsname}%
349     \expandafter\noexpand\csname the#1\endcsname}%
350   \@ifnextchar[%]
351   {\float@newx{#1}}%
352   {\@ifundefined{c@#1}{\newcounter{#1}\@namedef{the#1}{\arabic{#1}}}%
353   {}}}%
354 }%
355 \PackageInfo{scrhack}{redefining \string\listof}%
356 \renewcommand*\listof[2]{%
357   \@ifundefined{ext@#1}{\float@error{#1}}{%
358     \@ifundefined{l@#1}{\expandafter\let\csname l@#1\endcsname\l@figure
359     \@ifundefined{l@#1}{-%
360       \DeclareTOCStyleEntry[level=1,numwidth=2.3em,indent=1.5em]
361       {default}{#1}%
362     }{}%
363   }{}%
364   \listoftoc[#{2}]{\csname ext@#1\endcsname}%
365 }%
366 }%
367 \scr@ifexpected{\float@addtolists}{-%

```

```

368     \long\def\float@addtolists#1{%
369         \def\float@do##1{\addtocontents{##1}{#1}} \the\float@exts}%
370     }{%
371         \PackageInfo{scrhack}{undefining \string\float@addtolists}%
372         \let\float@addtolists\relax
373     }{%
374         \PackageWarningNoLine{scrhack}{unkown \string\float@addtolists\space
375             definition found!\MessageBreak
376             Maybe you are using a unsupported float version}%
377     }%
378 }{%
379     \PackageWarningNoLine{scrhack}{unknown \string\listof\space
380         definition found!\MessageBreak
381         Maybe you are using a unsupported float version}%
382 }%
383 }{%
384     \PackageWarningNoLine{scrhack}{unknown \string\newfloat\space
385         definition found!\MessageBreak
386         Maybe you are using a unsupported float version}%
387 }
388 </float & body>

```

2.6 The listings hack

Package `listings` 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.

```

389 <*package & option>
390 \KOMA@ifkey{listings}{@scrhack@listings}
391 \KOMAExecuteOptions{listings=true}
392 </package & option>
393 <*package & body>
394 \AfterPackage*{listings}{%
395     \@ifpackagelater{lstmisc}{2024/02/15}{%
396         \PackageInfo{scrhack}{listings hack deactivated because of\MessageBreak
397             detection of listings version, that doesn't\MessageBreak
398             need that hack,}%
399         \KOMAExecuteOptions[scrhack.sty]{listings=false}%
400     }{%
401         \if@scrhack@listings\scr@hack@load\@pkgextension{listings}\fi
402     }%
403     \KOMA@key[scrhack.sty]{listings}[true]{%

```

```

404   \PackageWarning{scrhack}{option 'listings=#1' ignored}%
405   \FamilyKeyStateProcessed
406 }%
407 }
408 </package & body>

```

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

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

```

409 (*listings & body)
410 \newcommand*{\scr@do@hack@listings}{%
411   \RequirePackage{tocbasic}%
412   \addtotoclist[float]{lol}%
413   \setuptoc{lol}{chapteratlist}%
414   \PackageInfo{scrhack}{redefining \string\lstlistoflistings}%
415   \renewcommand*{\lstlistoflistings}{\listoftoc[{\lstlistlistingname}]{lol}}%
416   \providecommand*{\ext@lstlisting}{lol}%
417   \@tempswafalse
418   \xpatchcmd\lst@MakeCaption
419     {\addcontentsline{lol}}%
420     {\addcontentsline{\ext@lstlisting}}%
421     {\PackageInfo{scrhack}{patching \string\lst@MakeCaption}\@tempswatrue}%
422     {\PackageWarning{scrhack}{cannot patch \string\lst@MakeCaption}}%
423   \@whiles\if@tempswa\fi{%
424     \xpatchcmd\lst@MakeCaption
425       {\addcontentsline{lol}}%
426       {\addcontentsline{\ext@lstlisting}}%
427       {}%
428       {\@tempswafalse}%
429   }%
430   \AtBeginDocument{%
431     \scr@ifexpected{\float@addtolists}{%
432       \def\float@addtolists##1{\addtocontents{lol}{##1}}%
433     }{%
434       \PackageInfo{scrhack}{undefining \string\float@addtolists}%
435       \let\float@addtolists\relax
436     }{%
437       \scr@ifexpected{\float@addtolists}{%
438         \def\float@addtolists##1{\addtocontents{lol}{##1}}%
439         \orig@float@addtolists{##1}%
440       }{%
441         \PackageInfo{scrhack}{setting \string\float@addtolists\MessageBreak
442           to \string\orig@float@addtolists}%
443         \let\float@addtolists\orig@float@addtolists
444       }{%
445         \PackageWarningNoLine{scrhack}{unkown \string\float@addtolists\space
446           definition found!\MessageBreak
447           Maybe you are using a unsupported listings version}%

```

```

448     }%
449     }%
450 }%

```

`\lst@makecaption` From version 3.30 setting of `\@capttype` 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`.

```

451 \expandafter\def\expandafter\lst@makecaption\expandafter{%
452 \expandafter\def\expandafter\@capttype\expandafter{%
453 \expandafter l\expandafter s\expandafter t\expandafter l%
454 \expandafter i\expandafter s\expandafter t\expandafter i%
455 \expandafter n\expandafter g\expandafter}\lst@makecaption
456 }%
457 \let\scr@do@hack@listings\relax
458 }
459 \scr@ifexpected{\lstlistoflistings}{%
460 \def\lstlistoflistings{\bgroup
461 \let\contentsname\lstlistlistingname
462 \let\lst@temp\@starttoc \def\@starttoc##1{\lst@temp{lol}}%
463 \tableofcontents \egroup}%
464 }{%
465 \scr@do@hack@listings
466 }{%
467 \scr@ifexpected{\lstlistoflistings}{%
468 \def\lstlistoflistings{%
469 \begingroup
470 \ifundefined{@restonecoltrue}{}{%
471 \iftwocolumn
472 \@restonecoltrue\onecolumn
473 \else
474 \@restonecolfalse
475 \fi
476 }%
477 \float@listhead{\lstlistlistingname}%
478 \parskip\z@\parindent\z@\parfillskip \z@ \@plus 1fil%
479 \@starttoc{lol}%
480 \ifundefined{@restonecoltrue}{}{%
481 \if@restonecol\twocolumn\fi
482 }%
483 \endgroup
484 }%
485 }{%
486 \scr@do@hack@listings
487 }{%
488 \PackageWarningNoLine{scrhack}{unknown \string\lstlistoflistings\space
489 definition found!\MessageBreak
490 Maybe you are using a unsupported listings version}%
491 }%
492 }
493 </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.)
494 (*package & option)
495 \RequirePackage{xpatch}%
496 \KOMA@ifkey{nomencl}{@scrhack@nomencl}
497 \KOMAExecuteOptions{nomencl=true}
498 \endpackage & option
499 (*package & body)
500 \AfterPackage*nomencl}{%
501   \KOMA@key[scrhack.sty]{nomencl}{%
502     \PackageWarning{scrhack}{option ‘nomencl’ ignored}%
503     \FamilyKeyStateProcessed
504   }%
505   \if@scrhack@nomencl\scr@hack@load\@pkgextension{nomencl}\fi
506 }
507 \endpackage & 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.

```
508 (*nomencl & body)
509 \RequirePackage{tocbasic}
```

With `nomencl` \geq 5.0 this should be all. Unfortunately it 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.

```
510 \@ifpackagelater{nomencl}{2019/01/01}{%
511   \PackageInfo{scrhack}{only ‘tocbasic’ loaded for ‘nomencl’}%
512   \expandafter\endinput
513 }-}
```

Otherwise we have to patch it.

```
514 \addtotoclist[nomencl]{nlo}
515 \addtotoclist[nomencl]{nls}
516 \xpatchcmd{\thenomenclature}{%
517   \@ifundefined{chapter}{%
518     {
519       \section*{\nomname}
520       \if@intoc\addcontentsline{toc}{section}{\nomname}\fi%
521     }%
522     {
523       \chapter*{\nomname}
```

```

524 \if@intoc\addcontentsline{toc}{chapter}{\nomname}\fi%
525 }%
526 }{%
527 \scr@hack@nomencl@head
528 }{%
529 \PackageInfo{scrhack}{\string\thenomenclature\space successfully patched}%
530 }{%

```

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

```

531 \xpatchcmd{\thenomenclature}{%
532 \@ifundefined{chapter}%
533 {
534 \section*{\nomname}
535 \if@intoc\addcontentsline{toc}{section}{\nomname}\fi%
536 }%
537 {
538 \chapter*{\nomname}
539 \markboth{\nomname}{\nomname}%
540 \if@intoc\addcontentsline{toc}{chapter}{\nomname}\fi%
541 }%
542 }{%
543 \scr@hack@nomencl@head
544 }{%
545 \PackageInfo{scrhack}{\string\thenomenclature\space successfully patched}%
546 }{%

```

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:

```

547 \xpatchcmd{\thenomenclature}{%
548 \@ifundefined{chapter}%
549 {
550 \section*{\nomname}
551 \if@intoc\addcontentsline{toc}{section}{\nomname}\fi%
552 }%
553 {
554 \chapter*{\nomname}
555 \@mkboth{\nomname}{\nomname}%
556 \if@intoc\addcontentsline{toc}{chapter}{\nomname}\fi%
557 }%
558 }{%
559 \scr@hack@nomencl@head
560 }{%
561 \PackageInfo{scrhack}{\string\thenomenclature\space successfully patched}%
562 }{%

```

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

```

563 \xpatchcmd{\thenomenclature}{%
564 \@ifundefined{chapter}%
565 }{%
566 \scr@hack@nomencl@head
567 \@gobbletwo
568 }{%

```

```

569     \PackageWarning{scrhack}{%
570         using emergency patch of \string\thenomenclature.\MessageBreak
571         This could happen if you are using an unknown\MessageBreak
572         release of package 'nomencl'.\MessageBreak
573         Mostly this patch does work too, but it also could\MessageBreak
574         break 'nomencl'. If this happens, please deactivate\MessageBreak
575         the 'nomencl' hack of 'scrhack'%
576     }%
577 }f%
578     \PackageWarning{scrhack}{%
579         cannot patch \string\thenomenclature.\MessageBreak
580         This could happen if you are using an unknown\MessageBreak
581         release of package 'nomencl'%
582     }%
583 }%
584 }%
585 }%
586 }
587 </nomencl & body>

```

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

```

588 (*nomencl & body)
589 \newcommand*\scr@hack@nomencl@head{%
590     \providecommand*\listofnlsname{\nomname}%
591     \def\@current{nls}%
592     \tocbasic@listhead{\nomname}%
593 }
594 </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.

```

595 (*nomencl & body)
596 \if@intoc
597     \setuptoc{nls}{totoc}
598 \fi
599 </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`, L^AT_EX 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:

$$10,95 \text{ pt} \cdot \frac{5}{13,6 \text{ pt}} \equiv 1,208$$

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 and the current font size. This is enough to have the same `\baselineskip` with both of the examples above.

`setspace` (*opt.*)

```

600 (*package & option)
601 \KOMA@ifkey{setspace}{@scrhack@setspace}
602 \KOMAExecuteOptions{setspace=true}
603 </package & option)
604 (*package & body)
605 \AfterPackage*{setspace}{%
606   \KOMA@key[scrhack.sty]{setspace}{%
607     \PackageWarning{scrhack}{option 'setspace' ignored}%
608     \FamilyKeyStateProcessed
609   }%
610   \if@scrhack@setspace\scr@hack@load\@pkgextension{setspace}\fi
611 }
612 </package & body)

```

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

```

613 (*setspace & body)
614 \scr@ifexpected{\onehalfspacing}{%
615   \long\def\onehalfspacing{%
616     \setstretch{1.25}% default
617     \ifcase \@ptsize \relax % 10pt

```

```

618     \setstretch {1.25}%
619     \or % 11pt
620     \setstretch {1.213}%
621     \or % 12pt
622     \setstretch {1.241}%
623     \fi
624   }%
625 }{%
626   \renewcommand*\onehalfspacing{%
627     \@tempdima=\dimexpr (\f@size pt)*1500/
628       (\dimexpr \f@baselineskip\relax)*\p@/1000\relax
629     \expandafter\setstretch\expandafter{\strip@pt\@tempdima}%
630   }%
631 }{%

632   \scr@ifexpected{\onehalfspacing}{%
633     \long\def\onehalfspacing{%
634       \setstretch{1.25}% default
635 % DPC 6.7b guard \@ptsize
636       \ifx\@ptsize@undefined\else
637         \ifcase \@ptsize \relax % 10pt
638           \setstretch {1.25}%
639           \or % 11pt
640           \setstretch {1.213}%
641           \or % 12pt
642           \setstretch {1.241}%
643         \fi
644       \fi
645     }%
646   }{%
647     \renewcommand*\onehalfspacing{%
648       \@tempdima=\dimexpr (\f@size pt)*1500/
649         (\dimexpr \f@baselineskip\relax)*\p@/1000\relax
650       \expandafter\setstretch\expandafter{\strip@pt\@tempdima}%
651     }%
652   }{%
653     \PackageWarning{scrhack}{unknown \string\onehalfspacing\space
654       definition found!\MessageBreak
655       Maybe you are using a unsupported setpace version}%
656   }%
657 }

```

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

```

658 \scr@ifexpected{\doublespacing}{%
659   \long\def\doublespacing{%
660     \setstretch{1.667}% default
661     \ifcase \@ptsize \relax % 10pt
662       \setstretch {1.667}%
663     \or % 11pt
664       \setstretch {1.618}%

```

```

665 \or % 12pt
666 \setstretch {1.655}%
667 \fi
668 }%
669 }{%
670 \renewcommand*{\doublespacing}{%
671 \@tempdima=\dimexpr (\f@size pt)*2000/
672 (\dimexpr \f@baselineskip\relax)*\p@/1000\relax
673 \expandafter\setstretch\expandafter{\strip@pt\@tempdima}%
674 }%
675 }{%

676 \scr@ifexpected{\doublespacing}{%
677 \long\def\doublespacing{%
678 \setstretch {1.667}% default
679 % DPC 6.7b guard \@ptsize
680 \ifx\@ptsize@undefined\else
681 \ifcase \@ptsize \relax % 10pt
682 \setstretch {1.667}%
683 \or % 11pt
684 \setstretch {1.618}%
685 \or % 12pt
686 \setstretch {1.655}%
687 \fi
688 \fi
689 }%
690 }{%
691 \renewcommand*{\doublespacing}{%
692 \@tempdima=\dimexpr (\f@size pt)*2000/
693 (\dimexpr \f@baselineskip\relax)*\p@/1000\relax
694 \expandafter\setstretch\expandafter{\strip@pt\@tempdima}%
695 }%
696 }{%
697 \PackageWarning{scrhack}{unknown \string\doublespacing\space
698 definition found!\MessageBreak
699 Maybe you are using a unsupported setpace version}%
700 }%
701 }
702 </setspace & body>

```

2.9 The lscape hack

The package does change `\textheight` inside `landscape` to the prior value of `\textwidth`, nevertheless it 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 `pdfscape`. The patches are done using `xpatch`.

`lscape (opt.)`

```

703 <*package & option>
704 \RequirePackage{xpatch}%
705 \KOMA@ifkey{landscape}{@scrhack@landscape}%
706 \KOMAEecuteOptions{landscape=true}%
707 </package & option>
708 <*package & body>
709 \AfterPackage*{landscape}{%
710   \if@scrhack@landscape\scrhack@load\@pkgextension{landscape}\else
711     \KOMA@key[scrhack.sty]{landscape}{%
712       \PackageWarning{scrhack}{option 'landscape' ignored}%
713       \FamilyKeyStateProcessed
714     }%
715   \fi
716 }
717 </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!

```

718 <*landscape & body>
719 \xpatchcmd{\landscape}{\textheight=\vsize}{%
720   \if@scrhack@landscape

```

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

```

721   \scrh@LT@textheight=\vsize
722   \let\scrh@LT@outputpage\@outputpage
723   \def\@outputpage{\scrh@LT@outputpage\global\@colht\scrh@LT@textheight}%
724   \else
725     \textheight=\vsize
726   \fi
727 }{%
728   \PackageInfo{scrhack}{\string\landscape\space patched to make
729     \string\textheight\space change optional}%
730 }{%
731   \PackageWarning{scrhack}{Cannot patch \string\landscape!\MessageBreak
732     Maybe you are using a unsupported landscape version}%
733   \@scrhack@landscapefalse
734 }

```

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

```

735 \newlength{\scrh@LT@textheight}
736 </landscape & 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 L^AT_EX 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.*)

```

737 <*package & option>
738 \KOMA@ifkey{standardsections}{@scrhack@standardsections}
739 \KOMAExecuteOptions{standardsections=false}
740 </package & option>
741 <*package & body>
742 \AtEndOfPackage{%
743   \if@scrhack@standardsections
744     \PackageInfo{scrhack}{loading standard classes sections hack}%
745     \input{standardsections.hak}%
746   \fi
747 }
748 </package & body>

```

`\part` The hack redefines all these commands to a definition copied from the standard L^AT_EX
`\chapter` classes version v1.4i. Copyright of this code is by the L^AT_EX3 Project. The complete,
`\section` unmodified code can be found on [https://ctan.org/tex-archive/macros/latex/base/](https://ctan.org/tex-archive/macros/latex/base/classes.dtx)
`\subsection` [classes.dtx](https://ctan.org/tex-archive/macros/latex/base/classes.dtx).

```

\subsubsection 749 <*standardsections & body>
\paragraph      750 \PackageWarningNoLine{scrhack}{%
\subparagraph   751   You are loosing every kind of support because of\MessageBreak
752   using the dangerous standard classes sections hack!\MessageBreak
753   This hack can break several classes and packages.\MessageBreak
754   However, sometimes it is needed to make a\MessageBreak
755   package work%
756 }%
757 \PackageInfo{scrhack}{%
758   Using code from the standard LaTeX classes.\MessageBreak
759   For more information about this code and the\MessageBreak
760   license of this code, please see:\MessageBreak
761   https://ctan.org/tex-archive/macros/latex/base/classes.dtx
762 }%
763 \@ifpackageloaded{hyperref}{%
764   \PackageWarning{scrhack}{this hack should not be loaded after 'hyperref'}%
765 }{}
766 \@ifpackageloaded{titlesec}{%
767   \PackageWarning{scrhack}{this hack should not be loaded after 'titlesec'}%
768 }{}
769 \@ifpackageloaded{sectsty}{%
770   \PackageWarning{scrhack}{this hack should not be loaded after 'sectsty'}%
771 }{}
772 \scr@ifundefinedorrelax{chapter}{%

```

It seems to be an article class, so we use the code from [article](#).

```

773   \scr@ifundefinedorrelax{c@part}{\newcounter{part}}{}%
774   \scr@ifundefinedorrelax{c@section}{\newcounter{section}}{}%

```

```

775 \scr@ifundefinedorrelax{c@subsection}{\newcounter{subsection}[section]}{}%
776 \scr@ifundefinedorrelax{c@subsubsection}{}%
777 \newcounter{subsubsection}[subsection]}{}%
778 \scr@ifundefinedorrelax{c@paragraph}{}%
779 \newcounter{paragraph}[subsubsection]}{}%
780 \scr@ifundefinedorrelax{c@subparagraph}{}%
781 \newcounter{subparagraph}[paragraph]}{}%
782 \renewcommand\thepart {\@Roman\c@part}%
783 \renewcommand\thesection {\@arabic\c@section}%
784 \renewcommand\thesubsection {\thesection.\@arabic\c@subsection}%
785 \renewcommand\thesubsubsection{\thesubsection.\@arabic\c@subsubsection}%
786 \renewcommand\theparagraph {\thesubsubsection.\@arabic\c@paragraph}%
787 \renewcommand\thesubparagraph {\theparagraph.\@arabic\c@subparagraph}%
788 \def\part{%
789 \if@noskipsec \leavevmode \fi
790 \par
791 \addvspace{4ex}%
792 \@afterindentfalse
793 \secdef\@part\@spart}%
794 \def\@part[#1]#2{%
795 \ifnum \c@secnumdepth >\m@ne
796 \refstepcounter{part}%
797 \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
798 \else
799 \addcontentsline{toc}{part}{#1}%
800 \fi
801 {\parindent \z@ \raggedright
802 \interlinepenalty \@M
803 \normalfont
804 \ifnum \c@secnumdepth >\m@ne
805 \Large\bfseries \partname\nobreakspace\thepart
806 \par\nobreak
807 \fi
808 \huge \bfseries #2%
809 \markboth{}{\par}%
810 \nobreak
811 \vskip 3ex
812 \@afterheading}%
813 \def\@spart#1{%
814 {\parindent \z@ \raggedright
815 \interlinepenalty \@M
816 \normalfont
817 \huge \bfseries #1\par}%
818 \nobreak
819 \vskip 3ex
820 \@afterheading}%
821 \def\section{\@startsection {section}{1}{\z@}%
822 {-3.5ex \@plus -1ex \@minus -.2ex}%
823 {2.3ex \@plus .2ex}%
824 {\normalfont\Large\bfseries}}%
825 \def\subsection{\@startsection{subsection}{2}{\z@}%
826 {-3.25ex\@plus -1ex \@minus -.2ex}%

```

```

827             {1.5ex \@plus .2ex}%
828             {\normalfont\large\bfseries}}%
829 \def\subsubsection{\@startsection{subsubsection}{3}{\z@}%
830             {-3.25ex\@plus -1ex \@minus -.2ex}%
831             {1.5ex \@plus .2ex}%
832             {\normalfont\normalsize\bfseries}}%
833 \def\paragraph{\@startsection{paragraph}{4}{\z@}%
834             {3.25ex \@plus 1ex \@minus .2ex}%
835             {-1em}%
836             {\normalfont\normalsize\bfseries}}%
837 \def\subparagraph{\@startsection{subparagraph}{5}{\parindent}%
838             {3.25ex \@plus 1ex \@minus .2ex}%
839             {-1em}%
840             {\normalfont\normalsize\bfseries}}%
841 }{%

```

It seems to be either a book or report class.

```

842 \scr@ifundefinedorrelax{c@part}{\newcounter{part}}{}%
843 \scr@ifundefinedorrelax{c@chapter}{\newcounter{chapter}}{}%
844 \scr@ifundefinedorrelax{c@section}{\newcounter{section}[chapter]}{}%
845 \scr@ifundefinedorrelax{c@subsection}{\newcounter{subsection}[section]}{}%
846 \scr@ifundefinedorrelax{c@subsubsection}{%
847     \newcounter{subsubsection}[subsection]}{}%
848 \scr@ifundefinedorrelax{c@paragraph}{%
849     \newcounter{paragraph}[subsubsection]}{}%
850 \scr@ifundefinedorrelax{c@subparagraph}{%
851     \newcounter{subparagraph}[paragraph]}{}%
852 \def\@chapapp{\chaptername}%
853 \scr@ifundefinedorrelax{frontmatter}{%

```

It seems not to be a book class.

```

854 \def\part{%
855     \if@openright
856         \cleardoublepage
857     \else
858         \clearpage
859     \fi
860     \thispagestyle{plain}%
861     \if@twocolumn
862         \onecolumn
863     \@tempwatrue
864 \else
865     \@tempwafalse
866 \fi
867     \null\vfil
868     \secdef\@part\@spart}
869
870 \def\@part[#1]#2{%
871     \ifnum \c@secnumdepth >-2\relax
872         \refstepcounter{part}%
873         \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
874     \else
875         \addcontentsline{toc}{part}{#1}%

```

```

876     \fi
877     \markboth{}{}%
878     {\centering
879     \interlinepenalty \@M
880     \normalfont
881     \ifnum \c@secnumdepth >-2\relax
882         \huge\bfseries \partname\nobreakspace\thepart
883         \par
884         \vskip 20\p@
885     \fi
886     \Huge \bfseries #2\par}%
887     \@endpart}
888 \def\@spart#1{%
889     {\centering
890     \interlinepenalty \@M
891     \normalfont
892     \Huge \bfseries #1\par}%
893     \@endpart}
894 \def\@endpart{\vfil\newpage
895             \if@twoside
896             \if@openright
897             \null
898             \thispagestyle{empty}%
899             \newpage
900             \fi
901             \fi
902             \if@tempswa
903             \twocolumn
904             \fi}
905 \def\chapter{\if@openright\cleardoublepage\else\clearpage\fi
906             \thispagestyle{plain}%
907             \global\@topnum\z@
908             \@afterindentfalse
909             \secdef\@chapter\@schapter}
910 \def\@chapter[#1]#2{\ifnum \c@secnumdepth >\m@ne
911             \refstepcounter{chapter}%
912             \typeout{\@chapapp\space\thechapter.}%
913             \addcontentsline{toc}{chapter}%
914                 {\protect\numberline{\thechapter}#1}%
915             \else
916             \addcontentsline{toc}{chapter}{#1}%
917             \fi
918             \chaptermark{#1}%
919             \addtocontents{lof}{\protect\addvspace{10\p@}}%
920             \addtocontents{lot}{\protect\addvspace{10\p@}}%
921             \if@twocolumn
922             \@topnewpage[\@makechapterhead{#2}]%
923             \else
924             \@makechapterhead{#2}%
925             \@afterheading
926             \fi}
927 \def\@makechapterhead#1{%

```



```

928     \vspace*{50\p@}%
929     {\parindent \z@ \raggedright \normalfont
930       \ifnum \c@secnumdepth >\m@ne
931         \huge\bfseries \@chapapp\space \thechapter
932         \par\nobreak
933         \vskip 20\p@
934       \fi
935       \interlinepenalty\@M
936       \Huge \bfseries #1\par\nobreak
937       \vskip 40\p@
938     }}
939 \def\@schapter#1{\if@twocolumn
940     \topnewpage[\@makeschapterhead{#1}]%
941     \else
942     \makeschapterhead{#1}%
943     \afterheading
944     \fi}
945 \def\@makeschapterhead#1{%
946     \vspace*{50\p@}%
947     {\parindent \z@ \raggedright
948       \normalfont
949       \interlinepenalty\@M
950       \Huge \bfseries #1\par\nobreak
951       \vskip 40\p@
952     }}
953 \def\section{\@startsection {section}{1}{\z@}%
954     {-3.5ex \@plus -1ex \@minus -.2ex}%
955     {2.3ex \@plus .2ex}%
956     {\normalfont\Large\bfseries}}%
957 \def\subsection{\@startsection{subsection}{2}{\z@}%
958     {-3.25ex\@plus -1ex \@minus -.2ex}%
959     {1.5ex \@plus .2ex}%
960     {\normalfont\large\bfseries}}%
961 \def\subsubsection{\@startsection{subsubsection}{3}{\z@}%
962     {-3.25ex\@plus -1ex \@minus -.2ex}%
963     {1.5ex \@plus .2ex}%
964     {\normalfont\normalsize\bfseries}}%
965 \def\paragraph{\@startsection{paragraph}{4}{\z@}%
966     {3.25ex \@plus 1ex \@minus .2ex}%
967     {-1em}%
968     {\normalfont\normalsize\bfseries}}%
969 \def\subparagraph{\@startsection{subparagraph}{5}{\parindent}%
970     {3.25ex \@plus 1ex \@minus .2ex}%
971     {-1em}%
972     {\normalfont\normalsize\bfseries}}%
973 }{%

```

It seems to be a book class.

```

974 \scr@ifundefinedorrelax{if@mainmatter}{%
975     \expandafter\newif\csname if@mainmatter\endcsname
976     }{}%
977 \def\frontmatter{%

```

```

978     \cleardoublepage
979     \@mainmatterfalse
980     \pagenumbering{roman}}%
981 \def\mainmatter{%
982     \cleardoublepage
983     \@mainmattertrue
984     \pagenumbering{arabic}}%
985 \def\backmatter{%
986     \ifopenright
987         \cleardoublepage
988     \else
989         \clearpage
990     \fi
991 \@mainmatterfalse}%
992 \def\part{%
993     \ifopenright
994         \cleardoublepage
995     \else
996         \clearpage
997     \fi
998     \thispagestyle{plain}%
999     \if@twocolumn
1000         \onecolumn
1001         \@tempwattrue
1002     \else
1003         \@tempwafalse
1004     \fi
1005     \null\vfil
1006     \secdef\@part\@spart}%
1007
1008 \def\@part[#1]#2{%
1009     \ifnum \c@secnumdepth >-2\relax
1010         \refstepcounter{part}%
1011         \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
1012     \else
1013         \addcontentsline{toc}{part}{#1}%
1014     \fi
1015     \markboth{}{}%
1016     {\centering
1017     \interlinepenalty \@M
1018     \normalfont
1019     \ifnum \c@secnumdepth >-2\relax
1020         \huge\bfseries \partname\nobreakspace\thepart
1021         \par
1022         \vskip 20\p@
1023     \fi
1024     \Huge \bfseries #2\par}%
1025 \@endpart}%
1026 \def\@spart#1{%
1027     {\centering
1028     \interlinepenalty \@M
1029     \normalfont

```

```

1030         \Huge \bfseries #1\par}%
1031     \endpart}%
1032 \def\@endpart{\vfil\newpage
1033         \if@twoside
1034         \if@openright
1035         \null
1036         \thispagestyle{empty}%
1037         \newpage
1038         \fi
1039         \fi
1040         \if@tempwa
1041         \twocolumn
1042         \fi}%
1043 \def\chapter{\if@openright\cleardoublepage\else\clearpage\fi
1044         \thispagestyle{plain}%
1045         \global\@topnum\z@
1046         \@afterindentfalse
1047         \secdef\@chapter\@schapter}%
1048 \def\@chapter[#1]#2{\ifnum \c@secnumdepth >\m@ne
1049         \if@mainmatter
1050         \refstepcounter{chapter}%
1051         \typeout{\@chapapp\space\thechapter.}%
1052         \addcontentsline{toc}{chapter}%
1053             {\protect\numberline{\thechapter}#1}%
1054         \else
1055         \addcontentsline{toc}{chapter}{#1}%
1056         \fi
1057         \else
1058         \addcontentsline{toc}{chapter}{#1}%
1059         \fi
1060         \chaptermark{#1}%
1061         \addtocontents{lof}{\protect\addvspace{10\p@}}%
1062         \addtocontents{lot}{\protect\addvspace{10\p@}}%
1063         \if@twocolumn
1064         \@topnewpage[\@makechapterhead{#2}]%
1065         \else
1066         \@makechapterhead{#2}%
1067         \@afterheading
1068         \fi}%
1069 \def\@makechapterhead#1{%
1070     \vspace*{50\p@}%
1071     {\parindent \z@ \raggedright \normalfont
1072     \ifnum \c@secnumdepth >\m@ne
1073         \if@mainmatter
1074         \huge\bfseries \@chapapp\space \thechapter
1075         \par\nobreak
1076         \vskip 20\p@
1077         \fi
1078         \fi
1079     \interlinepenalty\@M
1080     \Huge \bfseries #1\par\nobreak
1081     \vskip 40\p@

```

```

1082     }%
1083 \def\@chapter#1{\if@twocolumn
1084     \topnewpage[\@makeschapterhead{#1}]%
1085     \else
1086     \@makeschapterhead{#1}%
1087     \@afterheading
1088     \fi}%
1089 \def\@makeschapterhead#1{%
1090     \vspace*{50\p@}%
1091     {\parindent \z@ \raggedright
1092     \normalfont
1093     \interlinepenalty\@M
1094     \Huge \bfseries #1\par\nobreak
1095     \vskip 40\p@
1096     }%
1097 \def\section{\@startsection {section}{1}{\z@}%
1098     {-3.5ex \@plus -1ex \@minus -.2ex}%
1099     {2.3ex \@plus .2ex}%
1100     {\normalfont\Large\bfseries}}%
1101 \def\subsection{\@startsection{subsection}{2}{\z@}%
1102     {-3.25ex\@plus -1ex \@minus -.2ex}%
1103     {1.5ex \@plus .2ex}%
1104     {\normalfont\large\bfseries}}%
1105 \def\subsubsection{\@startsection{subsubsection}{3}{\z@}%
1106     {-3.25ex\@plus -1ex \@minus -.2ex}%
1107     {1.5ex \@plus .2ex}%
1108     {\normalfont\normalsize\bfseries}}%
1109 \def\paragraph{\@startsection{paragraph}{4}{\z@}%
1110     {3.25ex \@plus 1ex \@minus .2ex}%
1111     {-1em}%
1112     {\normalfont\normalsize\bfseries}}%
1113 \def\subparagraph{\@startsection{subparagraph}{5}{\parindent}%
1114     {3.25ex \@plus 1ex \@minus .2ex}%
1115     {-1em}%
1116     {\normalfont\normalsize\bfseries}}%
1117 }%
1118 }% End of the derived code.

```

`\@startsection` KOMA-Script classes redefine this internal L^AT_EX 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.

```

1119 \scr@ifundefinedorrelax{scr@saved@startsection}{}{%
1120   \ifx\@startsection\scr@saved@startsection\else
1121     \PackageWarningNoLine{scrhack}{%
1122       Resetting \string\@startsection,\MessageBreak
1123       because of loading standard classes sections hack%
1124     }%
1125     \let\@startsection\scr@saved@startsection
1126   \fi
1127 }

```

```

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

```

```

1128 \scr@ifundefinedorrelax{scr@latex@sect}{-}{%
1129   \ifx\@sect\scr@latex@sect\else
1130     \PackageWarningNoLine{scrhack}{%
1131       Resetting \string\@sect,\MessageBreak
1132       because of loading standard classes sections hack%
1133     }%
1134     \let\@sect\scr@latex@sect
1135   \fi
1136 }
1137 \scr@ifundefinedorrelax{scr@latex@ssect}{-}{%
1138   \ifx\@ssect\scr@latex@ssect\else
1139     \PackageWarningNoLine{scrhack}{%
1140       Resetting \string\@ssect,\MessageBreak
1141       because of loading standard classes sections hack%
1142     }%
1143     \let\@ssect\scr@latex@ssect
1144   \fi
1145 }
1146 \scr@ifundefinedorrelax{scr@latex@xsect}{-}{%
1147   \ifx\@xsect\scr@latex@xsect\else
1148     \PackageWarningNoLine{scrhack}{%
1149       Resetting \string\@xsect,\MessageBreak
1150       because of loading standard classes sections hack%
1151     }%
1152     \let\@xsect\scr@latex@xsect
1153   \fi
1154 }

```

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

```

\ProvideSectionCommand 1155 \scr@ifundefinedorrelax{DeclareSectionCommand}{-}{%
\DeclareSectionCommands 1156 \PackageWarningNoLine{scrhack}{%
\RedeclareSectionCommands 1157 \string\DeclareSectionCommand\space
\DeclareNewSectionCommands 1158 and all derivatives deactivated,\MessageBreak
\ProvideSectionCommands 1159 because of loading standard classes sections hack%
1160 }%
1161 \let\DeclareSectionCommand\undefined
1162 \let\DeclareSectionCommands\undefined
1163 \let\RedeclareSectionCommand\undefined
1164 \let\RedeclareSectionCommands\undefined
1165 \let\DeclareNewSectionCommand\undefined
1166 \let\DeclareNewSectionCommands\undefined
1167 \let\ProvideSectionCommand\undefined
1168 \let\ProvideSectionCommands\undefined
1169 }

```

\scr@chapter@before@hyperref@patch Now, definitions without respecting `hyperref` are used, so `hyperref` should again be able to
\scr@chapter@after@hyperref@patch patch them. So the hack should deactivate the patch deactivation of the KOMA-Script
classes.

```

1170 \let\scr@chapter@before@hyperref@patch\relax

```

```
1171 \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.

```
1172 \scr@ifundefinedorrelax{scr@osectarg}{-}{-%
1173   \def\reserved@a{0}%
1174   \ifx\scr@osectarg\reserved@a\else
1175     \let\scr@osectarg\reserved@a
1176     \PackageWarningNoLine{scrhack}{-%
1177       Extended option handling for section commands deactivated,\MessageBreak
1178       because of loading standard classes sections hack%
1179     }%
1180   \fi
1181 }%
```

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

```
1182 \scr@ifundefinedorrelax{scr@activate@xsection}{-}{-%
1183   \renewcommand*{\scr@activate@xsection}[1]{%
1184     \edef\scr@osectarg{#1}\def\reserved@a{0}%
1185     \ifx\scr@osectarg\reserved@a
1186       \else
1187         \let\scr@osectarg\reserved@a
1188         \PackageWarning{scrhack}{-%
1189           Because of loading the standard classes\MessageBreak
1190           sections hack, the extended option handling\MessageBreak
1191           of section commands failed%
1192         }%
1193       \fi
1194     }%
1195 }
```

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

```
1196 \scr@ifundefinedorrelax{partformat}{-}{-%
1197   \PackageWarningNoLine{scrhack}{-%
1198     Usage of \string\partformat\space
1199     deactivated,\MessageBreak
1200     because of loading the standard classes sections hack%
1201   }%
1202 }
1203 \scr@ifundefinedorrelax{chapterformat}{-}{-%
1204   \PackageWarningNoLine{scrhack}{-%
1205     Usage of \string\chapterformat\space
1206     deactivated,\MessageBreak
1207     because of loading the standard classes sections hack%
1208   }%
1209 }
```

```

\sectionformat All these commands have to be resetted to not use the automatic dot and to add a \quad
\subsectionformat instead of an \enskip.
\subsubsectionformat 1210 \scr@ifundefinedorrelax{sectionformat}{-}{%
\paragraphformat 1211 \PackageWarningNoLine{scrhack}{%
\subparagraphformat 1212 \string\sectionformat\space set to
1213 '\string\thesection\string\quad',\MessageBreak
1214 because of loading the standard classes sections hack%
1215 }%
1216 \renewcommand*{\sectionformat}{\thesection\quad}%
1217 }
1218 \scr@ifundefinedorrelax{subsectionformat}{-}{%
1219 \PackageWarningNoLine{scrhack}{%
1220 \string\subsectionformat\space set to
1221 '\string\thesubsection\string\quad'\MessageBreak
1222 because of loading the standard classes sections hack%
1223 }%
1224 \renewcommand*{\subsectionformat}{\thesubsection\quad}%
1225 }
1226 \scr@ifundefinedorrelax{subsubsectionformat}{-}{%
1227 \PackageWarningNoLine{scrhack}{%
1228 \string\subsubsectionformat\space set to
1229 '\string\thesubsubsection\string\quad'\MessageBreak
1230 because of loading the standard classes sections hack%
1231 }%
1232 \renewcommand*{\subsubsectionformat}{\thesubsubsection\quad}%
1233 }
1234 \scr@ifundefinedorrelax{paragraphformat}{-}{%
1235 \PackageWarningNoLine{scrhack}{%
1236 \string\paragraphformat\space set to
1237 '\string\theparagraph\string\quad'\MessageBreak
1238 because of loading the standard classes sections hack%
1239 }%
1240 \renewcommand*{\paragraphformat}{\theparagraph\quad}%
1241 }
1242 \scr@ifundefinedorrelax{subparagraphformat}{-}{%
1243 \PackageWarningNoLine{scrhack}{%
1244 \string\subparagraphformat\space set to
1245 '\string\thesubparagraph\string\quad'\MessageBreak
1246 because of loading the standard classes sections hack%
1247 }%
1248 \renewcommand*{\subparagraphformat}{\thesubparagraph\quad}%
1249 }

```

\partlineswithprefixformat These KOMA-Script commands are not used any longer.

```

\chapterlineswithprefixformat 1250 \scr@ifundefinedorrelax{partlineswithprefixformat}{-}{%
\chapterlinesformat 1251 \PackageWarningNoLine{scrhack}{%
\sectionlinesformat 1252 Usage of \string\partlineswithprefixformat\space deactivated,\MessageBreak
\sectioncatchphraseformat 1253 because of loading the standard classes sections hack%
1254 }%
1255 }
1256 \scr@ifundefinedorrelax{chapterlineswithprefixformat}{-}{%

```

```

1257 \PackageWarningNoLine{scrhack}{%
1258 Usage of \string\chapterlineswithprefixformat\space and\MessageBreak
1259 \string\chapterlinesformat\space deactivated,\MessageBreak
1260 because of using standard classes sections hack%
1261 }%
1262 }
1263 \scr@ifundefinedorrelax{sectionlinesformat}{}{-%
1264 \PackageWarningNoLine{scrhack}{%
1265 Usage of \string\sectionlinesformat\space and\MessageBreak
1266 \string\sectioncatchphraseformat\space deactivated,\MessageBreak
1267 because of using standard classes sections hack%
1268 }%
1269 }

```

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

```

1270 \scr@ifundefinedorrelax{KOMAClassName}{}{-%
1271 \PackageWarningNoLine{scrhack}{%
1272 KOMA-Script option ‘headings’ removed,\MessageBreak
1273 because of loading standard classes sections hack%
1274 }%
1275 \RelaxFamilyKey[.\KOMAClassFileName]{KOMA}{headings}%
1276 \KOMA@kav@removekey{\KOMAClassFileName}{headings}%
1277 }

```

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

```

1278 \def\scr@class@titlesec@warning{%
1279 \PackageWarning{scrhack}{%
1280 Usage of ‘titlesec’ should work, because of\MessageBreak
1281 loading the standard classes sections hack.\MessageBreak
1282 However, you should know, there is not any\MessageBreak
1283 support for this combination by the\MessageBreak
1284 KOMA-Script author%
1285 }%
1286 }
1287 </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](#).

```

1288 <*package & option>
1289 \KOMAProcessOptions\relax
1290 </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.

C		<code>\sectioncatchphraseformat nomencl (<i>opt.</i>)</code> 494
<code>\chapter</code> 749	<code>\sectionformat</code> 1250	
<code>\chapterformat</code> 1196	<code>\sectionlinesformat</code> 1210	O
<code>\chapterlinesformat</code> 1250	<code>\subparagraph</code> 1250	<code>\onehalfspacing</code> 613
<code>\chapterlineswithprefixformat</code> 1250	<code>\subparagraphformat</code> 749	Options:
Commands:	<code>\subsection</code> 1210	<code>float</code> 285
<code>\chapter</code> 749	<code>\subsectionformat</code> 749	<code>floatrow</code> 192
<code>\chapterformat</code> 1196	<code>\subsubsection</code> 1210	<code>headings</code> 1270
<code>\chapterlinesformat</code> 1250	<code>\subsubsectionformat</code> 749	<code>hyperref</code> 58
<code>\chapterlineswithprefixformat</code> 1250	<code>\thenomenclature</code> 508	<code>listings</code> 389
<code>\DeclareNewFloatType</code> 207		<code>lscape</code> 703
<code>\DeclareNewSectionCommand</code> 1155	D	<code>nomencl</code> 494
<code>\DeclareNewSectionCommands</code> 1155	<code>\DeclareNewFloatType</code> 207	<code>setspace</code> 600
<code>\DeclareSectionCommand</code> 1155	<code>\DeclareNewSectionCommand</code> 1155	<code>standardsections</code> 737
<code>\DeclareSectionCommands</code> 1155	<code>\DeclareNewSectionCommands</code> 1155	
<code>\doublespacing</code> 658	<code>\DeclareSectionCommand</code> 1155	P
<code>\landscape</code> 718	<code>\DeclareSectionCommands</code> 1155	<code>\paragraph</code> 749
<code>\listof</code> 207 , 298	<code>\DeclareSectionCommands</code> 1155	<code>\paragraphformat</code> 1210
<code>\lstlistoflistings</code> 409	<code>\doublespacing</code> 658	<code>\part</code> 749
<code>\newfloat</code> 298		<code>\partformat</code> 1196
<code>\onehalfspacing</code> 613	F	<code>\partlineswithprefixformat</code> 1250
<code>\paragraph</code> 749	<code>float (<i>opt.</i>)</code> 285	<code>\ProvideSectionCommand</code> 1155
<code>\paragraphformat</code> 1210	<code>floatrow (<i>opt.</i>)</code> 192	<code>\ProvideSectionCommands</code> 1155
<code>\part</code> 749	H	
<code>\partformat</code> 1196	<code>headings (<i>opt.</i>)</code> 1270	R
<code>\partlineswithprefixformat</code> 1250	<code>hyperref (<i>opt.</i>)</code> 58	<code>\RedeclareSectionCommand</code> 1155
<code>\ProvideSectionCommand</code> 1155		<code>\RedeclareSectionCommands</code> 1155
<code>\ProvideSectionCommands</code> 1155	L	
<code>\RedeclareSectionCommand</code> 1155	<code>\landscape</code> 718	S
<code>\RedeclareSectionCommands</code> 1155	Lengths (internal):	<code>\section</code> 749
<code>\section</code> 749	<code>\scrh@LT@textheight</code> 735	<code>\sectioncatchphraseformat</code> 1250
	<code>listings (<i>opt.</i>)</code> 389	<code>\sectionformat</code> 1210
	<code>\listof</code> 207 , 298	<code>\sectionlinesformat</code> 1250
	<code>lscape (<i>opt.</i>)</code> 703	<code>setspace (<i>opt.</i>)</code> 600
	<code>\lstlistoflistings</code> 409	<code>standardsections (<i>opt.</i>)</code> 737
	<code>\newfloat</code> 298	<code>\subparagraph</code> 749
		<code>\subparagraphformat</code> 1210
	N	<code>\subsection</code> 749
		<code>\subsectionformat</code> 1210
		<code>\subsubsection</code> 749

<code>\subsubsectionformat</code>	1210	<code>\ext@lstlisting</code>	409	<code>\scr@class@titlesec@warning</code>	1170
T					
TeX macros (internal):					
<code>\@outputpage</code>	721	<code>\float@addtolists</code>	207, 298, 409	<code>\scr@do@hack@listings</code>	1278
<code>\@schapter</code>	90	<code>\if@intoc</code>	595	<code>\scr@do@hack@listings</code>	409
<code>\@sect</code>	1128	<code>\lst@MakeCaption</code>	409	<code>\scr@hack@load</code>	37
<code>\@spart</code>	90	<code>\lst@makecaption</code>	451	<code>\scr@hack@load</code>	37
<code>\@ssect</code>	90, 1128	<code>\scr@activate@xsection</code>	1182	<code>\scr@hack@nomenc@head</code>	588
<code>\@startsection</code>	1119	<code>\scr@chapter@after@hyperref@patch@ifexpected</code>	26	<code>\scr@osectarg</code>	1172
<code>\@xsect</code>	1128	<code>\scr@chapter@before@hyperref@patch@ifexpected</code>	1170	<code>\scr@osectarg</code>	1172
		<code>\scr@chapter@before@hyperref@patch@ifexpected</code>	508	<code>\scr@osectarg</code>	1172

Change History

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

v3.31 – 2020/07/08		<code>listings</code> : initial dot in member argument of <code>\KOMA@key</code> removed . . .	11
	<code>\scr@hack@nomenc1@head</code> : internal		
	<code>\list@fname</code> eliminated		16
v3.34 – 2021/06/03	<code>floatrow</code> : deactivation of the <code>float</code> hack		6
v3.36 – 2022/02/08	General: require package <code>scrlogo</code> instead of defining <code>\KOMAScript</code>		1
	switch over from <code>scrdoc</code> to <code>koma-script-source-doc</code>		1
	whole implementation documentation in English		1
v3.39 – 2022/11/11	<code>float</code> : initial dot in member argument of <code>\KOMA@key</code> removed		9
	<code>floatrow</code> : initial dot in member argument of <code>\KOMA@key</code> removed . . .		6
	<code>headings</code> : initial dot in member argument of option storage commands removed		32
	<code>hyperref</code> : initial dot in member argument of <code>\KOMA@key</code> and <code>\KOMAExecuteOptions</code> removed		3
		<code>listings</code> : initial dot in member argument of <code>\KOMA@key</code> removed . . .	11
		<code>lscape</code> : initial dot in member argument of <code>\KOMA@key</code> removed	19
		<code>nomenc1</code> : initial dot in member argument of <code>\KOMA@key</code> removed . . .	14
		<code>setspace</code> : initial dot in member argument of <code>\KOMA@key</code> removed . . .	17
v3.39 – 2023/03/15	<code>\doublespacing</code> : support for <code>setspace</code> 6.7b added		19
	<code>\onehalfspacing</code> : support for <code>setspace</code> 6.7b added		18
v3.40 – 2023/04/17	General: guide names changed		1
v3.41 – 2023/06/28	<code>\scr@do@hack@listings</code> : patching <code>\lst@MakeCaption</code> to use <code>\ext@lstlisting</code>		12
v3.42 – 2024/04/17	<code>listings</code> : automatic deactivation for new <code>listings</code>		11