

KOMA-Script File `scrhack.dtx` *

Markus Kohm[†]

package

Some packages from other authors may have problems with KOMA-Script. In my opinion some packages could be improved. With some packages this makes only sense, if KOMA-Script was used. With some other packages the package author has another opinion. Sometimes proposals was never answered. Package `scrhack` contains all those improvement proposals for other packages. This means, `scrhack` redefines macros of packages from other authors! The redefinitions are only activated, if those packages were loaded. Users may prevent `scrhack` from redefining macros of individual packages.

Contents

1	The hyperref hack	2
2	The float hack	2
3	The floatrow hack	3
4	The listings hack	4
5	The setspace hack	4
6	Implementation of <code>scrhack</code>	5
6.1	Optionen	5
6.2	Verwendete Anweisungen	5
6.3	Der hyperref-Hack	6
6.4	Der float-Hack	9
6.5	Der floatrow-Hack	12

*This file is version (hacking of `scrhack.dtx`.

[†][mailto:komascript\(at\)gmh.info](mailto:komascript(at)gmh.info)

6.6	Der listings-Hack	14
6.7	Der setspace-Hack	17
6.8	Optionen ausführen	19

1 The hyperref hack

Before version 6.79h package `hyperref` does behave different at part, chapter, and section headings that get no number. If they get no number, because of to low counter

`secnumdepth` `hyperref` sets an anchor for links and bookmarks before the heading. Same would be, if the headings have a number. But if the headings get no number because of usage of the star version of the commands, e.g., `\part*`, `\chapter*` or `\section*`, the anchor for links and bookmarks are set after the headings. The anchors for numbered headings are always set before the headings.

Package `scrhack` redefines some macros of some `hyperref` driver files, e.g., `hpdftex.def`, after loading the `hyperref` driver file. With this redefinitions the anchor of not numbered headings will be set always before the headings, too.

You may switch off the `hyperref` hack loading package `scrhack` with option `hyperref=false`. You may also switch off the `hyperref` hack using `\KOMAOptions{hyperref=false}` or `\KOMAoption{hyperref}{false}` somewhere after loading package `scrhack`, but before loading the `hyperref` driver package, that is by default after loading the package.

2 The float hack

Package `float` uses macros `\float@listhead` to set the headings of a float listing and `\float@addtolists` to add informations to all float listings. These macros where proposed by the `KOMA-Script` author for some years. In theory those macros may be used by several class and package authors to deligate some parts of the creation of a float listing to the class. This would increase the compatibility of packages and classes. But unfortunately some package authors, even the author of package `float`, implemented the commands in such a way, that these packages will become incompatible to each other.

Because of this KOMA-Script stopped support for `\float@addtolists` and `\float@listhead` with version 3. Instead of this KOMA-Script supports several improvements for package authors using KOMA-Script package `tocbasic`.

Package `scrhack` redefines some macros of package `float` to not longer use `\float@addtolists` and `\float@listhead` but use the interface of package `tocbasic`. This does not only improve the compatibility of KOMA-Script and package `float`, but also improves the compatibility of packages `babel` and `float`.

You may switch off the float hack loading package `scrhack` with option `float=false`. You may also switch off the float hack using `\KOMAOPTIONS{float=false}` or `\KOMAoption{float}{false}` somewhere after loading package `scrhack`, but before loading package `float`.

3 The floatrow hack

Package `floatrow` uses macros `\float@listhead` to set the headings of a float listing and `\float@addtolists` to add informations to all float listings. These macros were proposed by the KOMA-Script author for some years. In theory those macros may be used by several class and package authors to delegate some parts of the creation of a float listing to the class. This would increase the compatibility of packages and classes. But unfortunately some package authors, even the author of package `floatrow`, implemented the commands in such a way, that these packages will become incompatible to each other.

Because of this KOMA-Script stopped support for `\float@addtolists` and `\float@listhead` with version 3. Instead of this KOMA-Script supports several improvements for package authors using KOMA-Script package `tocbasic`.

Package `scrhack` redefines some macros of package `floatrow` to not longer use `\float@addtolists` and `\float@listhead` but use the interface of package `tocbasic`. This does not only improve the compatibility of KOMA-Script and package `floatrow`, but also improves the compatibility of packages `babel` and `floatrow`.

You may switch off the floatrow hack loading package `scrhack` with option `floatrow=false`. You may also switch off the floatrow hack using `\KOMAOPTIONS{floatrow=false}` or `\KOMAoption{floatrow}{false}` somewhere after loading package `scrhack`, but before loading package `floatrow`.

4 The listings hack

Package `listings` uses macros `\float@listhead` to set the headings of a float listing, if defined, and `\float@addtolists` to add informations to all float listings. These macros were proposed by the KOMA-Script author for some years. In theory those macros may be used by several class and package authors to delegate some parts of the creation of a float listing to the class. This would increase the compatibility of packages and classes. But unfortunately some package authors, even the author of package `float`, implemented the commands in such a way, that these packages may become incompatible to each other.

Because of this KOMA-Script stopped support for `\float@addtolists` and `\float@listhead` with version 3. Instead of this KOMA-Script supports several improvements for package authors using KOMA-Script package `tocbasic`.

Package `scrhack` redefines some macros of package `listings` to not longer use `\float@addtolists` and `\float@listhead` but use the interface of package `tocbasic`. This does not only improve the compatibility of KOMA-Script and package `listings`, but also improves the compatibility of packages `babel` and `listings`.

Note: A significant change with `scrhack` is, that KOMA-Script options like `lists=totoc` or `lists=totocnumbered` does only change the behaviour of `\listoflistings`, if they are set after loading package `listings`.

You may switch off the listings hack loading package `scrhack` with option `listings=false`. You may also switch off the listings hack using `\KOMAoptions{listings=false}` or `\KOMAoption{listings}{false}` somewhere after loading package `scrhack`, but before loading package `listings`.

5 The setspace hack

Package `setspace` defines macros `\onehalfspacing` and `\doublespacing` using `\@ptsize` as an argument of `\ifcase`. But if `\@ptsize` is not an integer but a real number, this fails, because the digits from the decimal points are interpreted as text of that case. Several solutions for this are thinkable. I've decided to redefine `\onehalfspacing` and `\doublespacing`. The new definition is more general and somehow more exact.

You can switch of the `setspace` hack loading package `scrhack` with option `setspace=false`. You may also switch of the `setspace` hack using `\KOMAOptions{setspace=false}` or `\KOMAOption{setspace}{false}` somewhere after loading package `scrhack`, but before loading package `setspace`.

Note: If you want to use `setspace` with package option `onehalfspacing` or `doublespacing` you have to load `scrhack` before `setspace`.

6 Implementation of `scrhack`

6.1 Optionen

Das Paket bedient sich `\KOMAOptions` etc. aus `scrkbase` (dieses wird übrigens direkt per `scrkbase.dtx` geladen).

Per Option kann gewählt werden, welche Manipulationen geladen werden sollen. Alle diese Optionen können jedoch nur bis zum Laden des entsprechenden Pakets oder dem Laden von `scrhack` gesetzt werden (es zählt, was später kommt). Anschließend sind sie wirkungslos.

6.2 Verwendete Anweisungen

`\scr@ifexpected` Wenn die im ersten Argument angegebene Anweisung nach Ausführung der im zweiten Argument angegebenen Anweisungen unverändert ist, dann soll das dritte Argument ausgeführt werden, sonst das vierte.

```

1 \newcommand{\scr@ifexpected}[2]{%
2   \begingroup
3     \let\@tempa#1
4     #2
5     \ifx\@tempa#1
6       \aftergroup\@firstoftwo
7     \else
8       \aftergroup\@secondoftwo
9     \fi
10  \endgroup
11 }
```

`\scr@hack@load` Wenn die Datei mit dem Namen des zweiten Arguments und der Endung des ersten Arguments so geladen wurde, dass \LaTeX eine Versionsinfo dazu gespeichert hat, dann soll zusätzlich der entsprechende Hack geladen werden.

```

12 \newcommand*{\scr@hack@load}[2]{%
13   \expandafter\ifx\csname ver@#2.#1\endcsname\relax
14     \expandafter\@secondoftwo
15   \else
16     \expandafter\@firstoftwo
17   \fi
18   {%
19     \PackageInfo{scrhack}{loading #2 hack}%
20     \edef\reserved@a{%
21       \noexpand\makeatletter\noexpand\input{#2.hak}%
22       \noexpand\catcode'\noexpand\@the\catcode'\@relax
23     }\reserved@a
24   }{%
25     \PackageInfo{scrhack}{ignoring #2 hack}%
26   }%
27 }

```

6.3 Der hyperref-Hack

hyperref setzt den Anker zu der Stern-Variante einer Überschrift hinter die Überschrift, während es bei der nicht Stern-Variante den Anker auch dann vor die Überschrift setzt, wenn die Überschrift aufgrund von `secnumdepth` nicht nummeriert wird. Der Hack setzt den Anker einheitlich vor die Überschrift.

hyperref

```

28 <*package & option>
29 \KOMA@ifkey{hyperref}{\scrhack@hyperref}%
30 \KOMAEecuteOptions{hyperref=true}%
31 </package & option>
32 <*package & body>

```

Hier muss ein wenig trickreicher gearbeitet werden, weil hyperref die Treiberdatei per `\AtEndOfPackage` lädt und der Hack erst danach installiert werden darf. Mit `\AfterPackage*` alleine, würde der Hack aber vor dem Laden der Treiberdatei installiert. Dafür können wir aber sicher sein, dass ein innerhalb von `\AfterPackage*` aufgerufenes `\AtEndOfPackage` garantiert nach dem Laden der Treiberdatei ausgeführt wird. Das funktioniert auch noch, wenn hyperref bereits geladen wurde. In dem Fall wird der Code einfach nach dem Ende von scrhack statt nach dem Ende von hyperref ausgeführt.

```

33 \AfterPackage*{hyperref}{%
34   \@ifpackagelater{hyperref}{2009/11/24}{%
35     \PackageInfo{scrhack}{hyperref hack deactivated because of\MessageBreak

```

```

36      detection of hyperref version, that doesn't\MessageBreak
37      need that hack,}%
38      \KOMAEecuteOptions[.scrhack.sty]{hyperref=false}%
39  }{%
40      \AtEndOfPackage{%
41          \KOMA@key[.scrhack.sty]{hyperref}{%
42              \PackageWarning{scrhack}{option 'hyperref=#1' ignored}%
43              \FamilyKeyStateProcessed
44          }%
45          \if@scrhack@hyperref\scr@hack@load\@pkgextension{hyperref}\fi
46      }%
47  }%
48 }
49 </package & body>

```

\@schapter Eigentlich wird hier gar nicht `hyperref.sty` verändert, sondern diverse
\@spart Treiberdateien. Sobald das Paket `hyperref` geladen ist, ist auch die passen-
\@ssect de Treiberdatei geladen und außerdem sind alle Treiberdateien, die ent-
sprechende Definitionen vornehmen, gleichermaßen betroffen. Also kann
der entsprechende Patch einfach erfolgen, wenn `hyperref` geladen ist (was
bereits von `\scr@hack@load` getestet wurde). Es muss also nur noch sicher-
gestellt werden, dass die umzudefinierenden Macros derzeit den erwarteten
Inhalt haben.

```

50 <*hyperref & body>
51 \scr@ifexpected\@schapter{%
52     \def\@schapter#1{%
53         \H@old@schapter{#1}%
54         \begingroup
55             \let\@mkboth\@gobbletwo
56             \Hy@GlobalStepCount\Hy@linkcounter
57             \xdef\@currentHref{\Hy@chapapp*.\the\Hy@linkcounter}%
58             \Hy@raisedlink{%
59                 \hyper@anchorstart{\@currentHref}\hyper@anchorend
60             }%
61         \endgroup
62     }%
63 }{%
64     \PackageInfo{scrhack}{redefining \string\@schapter}%
65     \def\@schapter#1{%
66         \begingroup
67             \let\@mkboth\@gobbletwo
68             \Hy@GlobalStepCount\Hy@linkcounter
69             \xdef\@currentHref{\Hy@chapapp*.\the\Hy@linkcounter}%
70             \Hy@raisedlink{%
71                 \hyper@anchorstart{\@currentHref}\hyper@anchorend
72             }%
73         \endgroup

```

```

74     \H@old@schapter{#1}%
75 }%
76 }{%
77 \scr@ifexpected\@schapter{%
78   \def\@schapter#1{%
79     \begingroup
80     \let\@mkboth\@gobbletwo
81     \Hy@GlobalStepCount\Hy@linkcounter
82     \xdef\@currentHref{\Hy@chapapp*.\the\Hy@linkcounter}%
83     \Hy@raisedlink{%
84       \hyper@anchorstart{\@currentHref}\hyper@anchorend
85     }%
86   \endgroup
87   \H@old@schapter{#1}%
88 }%
89 }{}{%
90   \PackageWarningNoLine{scrhack}{unknown \string\@schapter\space
91     definition found!\MessageBreak
92     Maybe you are using a unsupported hyperref version}%
93 }%
94 }
95
96 \scr@ifexpected\@spart{%
97   \def\@spart#1{%
98     \H@old@spart{#1}%
99     \Hy@GlobalStepCount\Hy@linkcounter
100    \xdef\@currentHref{part*.\the\Hy@linkcounter}%
101    \Hy@raisedlink{%
102      \hyper@anchorstart{\@currentHref}\hyper@anchorend
103    }%
104  }%
105 }{%
106   \PackageInfo{scrhack}{redefining \string\@spart}%
107   \def\@spart#1{%
108     \Hy@GlobalStepCount\Hy@linkcounter
109     \xdef\@currentHref{part*.\the\Hy@linkcounter}%
110     \Hy@raisedlink{%
111       \hyper@anchorstart{\@currentHref}\hyper@anchorend
112     }%
113     \H@old@spart{#1}%
114   }%
115 }{}%
116 \scr@ifexpected\@spart{%
117   \def\@spart#1{%
118     \Hy@GlobalStepCount\Hy@linkcounter
119     \xdef\@currentHref{part*.\the\Hy@linkcounter}%
120     \Hy@raisedlink{%
121       \hyper@anchorstart{\@currentHref}\hyper@anchorend
122     }%

```



```

123     \H@old@spart{#1}%
124   }%
125 }{}{%
126   \PackageWarningNoLine{scrhack}{unknown \string\@spart\space
127     definition found!\MessageBreak
128     Maybe you are using a unsupported hyperref version}%
129 }%
130 }
131
132 \scr@ifexpected\@ssect{%
133   \def\@ssect#1#2#3#4#5{%
134     \H@old@ssect{#1}{#2}{#3}{#4}{#5}%
135     \phantomsection
136   }%
137 }{%
138   \PackageInfo{scrhack}{redefining \string\@ssect}%
139   \def\@ssect#1#2#3#4#5{%
140     \H@old@ssect{#1}{#2}{#3}{#4}{\phantomsection\ignorespaces#5}%
141   }%
142 }{%
143   \scr@ifexpected\@ssect{%
144     \def\@ssect#1#2#3#4#5{%
145       \H@old@ssect{#1}{#2}{#3}{#4}{\phantomsection\ignorespaces#5}%
146     }%
147   }{}{%
148     \PackageWarningNoLine{scrhack}{unknown \string\@ssect\space
149       definition found!\MessageBreak
150       Maybe you are using a unsupported hyperref version}%
151   }%
152 }
153 </hyperref & body>

```

6.4 Der float-Hack

Das float-Paket verwendet das Makro `\float@listhead` zum Setzen der Überschriften. Dies wird seit KOMA-Script 3 nicht mehr empfohlen und fliegt demnächst komplett aus der Unterstützung. Stattdessen wird empfohlen, dass Pakete `tocbasic` unterstützen. Der Aufwand dafür ist sehr gering und wird mit vielen neuen Möglichkeiten belohnt.

Dieser Hack rüstet die `tocbasic`-Unterstützung für float nach.

float

```

154 <*package & option>
155 \KOMA@ifkey{float}{@scrhack@float}%
156 \KOMAExecuteOptions{float=true}%

```

```

157 </package & option>
158 <*package & body>
159 \AfterPackage*{float}{%
160   \KOMA@key[.scrhack.sty]{float}{%
161     \PackageWarning{scrhack}{option 'float' ignored}%
162     \FamilyKeyStateProcessed
163   }%
164   \if@scrhack@float\scr@hack@load\@pkgextension{float}\fi
165 }
166 </package & body>

```

`\newfloat` Über die Anweisung `\newfloat` wird eine neue Gleitumgebung definiert. Hier muss die neue Erweiterung aus dem dritten Argument `tocbasic` bekannt gemacht werden.

`\listof` Über die Anweisung `\listof` wird ein Verzeichnis für Gleitumgebungen ausgegeben. Hier muss schlicht die entsprechende Anweisung von `tocbasic` verwendet werden.

`\float@addtolists` Diese Anweisung wird nicht länger benötigt und daher auf die ursprüngliche Definition zurückgesetzt.

```

167 <*float & body>
168 \scr@ifexpected{\newfloat}{%
169   \long\def\newfloat#1#2#3{\@namedef{ext@#1}{#3}
170     \let\float@do=\relax
171     \xdef\@tempa{\noexpand\float@exts{\the\float@exts \float@do{#3}}}%
172     \@tempa
173     \floatplacement{#1}{#2}%
174     \@ifundefined{fname@#1}{\floatname{#1}{#1}}{}
175     \expandafter\edef\csname ftype@#1\endcsname{\value{float@type}}%
176     \addtocounter{float@type}{\value{float@type}}
177     \restylefloat{#1}%
178     \expandafter\edef\csname fnum@#1\endcsname%
179     {\expandafter\noexpand\csname fname@#1\endcsname{}
180       \expandafter\noexpand\csname the#1\endcsname}
181     \@ifnextchar[%]
182     {\float@newx{#1}}%
183     {\@ifundefined{c@#1}{\newcounter{#1}\@namedef{the#1}{\arabic{#1}}}%
184       {}}}%
185 }{%
186   \scr@ifexpected{\listof}{%
187     \def\listof#1#2{%
188       \@ifundefined{ext@#1}{\float@error{#1}}{%
189         \@namedef{l@#1}{\@dottedtocline{1}{1.5em}{2.3em}}%
190         \float@listhead{#2}%
191         \begingroup\setlength{\parskip}{\z@}%
192         \@starttoc{\@nameuse{ext@#1}}%

```

```

193         \endgroup}}}%
194     }{%
195         \RequirePackage{tocbasic}%
196         \PackageInfo{scrhack}{redefining \string\newfloat}%
197         \renewcommand\newfloat[3]{%
198             \ifattoclist{#3}{%
199                 \PackageError{scrhack}{extension ‘#3’ already in use}{%
200                     Each extension may be used only once.\MessageBreak
201                     You, the class, or another package already uses extension
202                     ‘#3’.\MessageBreak
203                     \string\newfloat\space command will be ignored!}%
204             }{%
205                 \addtotoclist[float]{#3}%
206                 \setuptoc{#3}{chapteratlist}%
207                 \@namedef{ext@#1}{#3}%
208                 \let\float@do=\relax
209                 \xdef\@tempa{\noexpand\float@exts{\the\float@exts \float@do{#3}}}%
210                 \@tempa
211                 \floatplacement{#1}{#2}%
212                 \@ifundefined{fname@#1}{\floatname{#1}{#1}}{}%
213                 \expandafter\edef\csname ftype@#1\endcsname{\value{float@type}}%
214                 \addtocounter{float@type}{\value{float@type}}
215                 \restylefloat{#1}%
216                 \expandafter\edef\csname fnum@#1\endcsname%
217                 {\expandafter\noexpand\csname fname@#1\endcsname{}
218                  \expandafter\noexpand\csname the#1\endcsname}%
219                 \@ifnextchar[%]
220                 {\float@newx{#1}}%
221                 {\@ifundefined{c@#1}{\newcounter{#1}\@namedef{the#1}{\arabic{#1}}}%
222                  {}}}}%
223     }%
224     \PackageInfo{scrhack}{redefining \string\listof}%
225     \renewcommand*\listof[2]{%
226         \@ifundefined{ext@#1}{\float@error{#1}}{%
227             \@ifundefined{l@#1}{\expandafter\let\csname l@#1\endcsname\l@figure
228              \@ifundefined{l@#1}{%
229                  \@namedef{l@#1}{\@dottedtocline{1}{1.5em}{2.3em}}}%
230             }{}%
231             \listoftoc[{#2}]{\csname ext@#1\endcsname}%
232         }%
233     }%
234     \scr@ifexpected{\float@addtolists}{%
235         \long\def\float@addtolists#1{%
236             \def\float@do##1{\addtocontents{##1}{#1}} \the\float@exts}%
237     }{%
238         \PackageInfo{scrhack}{undefining \string\float@addtolists}%
239         \let\float@addtolists\relax
240     }{%
241         \PackageWarningNoLine{scrhack}{unkown \string\float@addtolists\space

```

```

242         definition found!\MessageBreak
243         Maybe you are using a unsupported float version}%
244     }%
245 }{%
246     \PackageWarningNoLine{scrhack}{unknown \string\listof\space
247         definition found!\MessageBreak
248         Maybe you are using a unsupported float version}%
249 }%
250 }{%
251     \PackageWarningNoLine{scrhack}{unknown \string\newfloat\space
252         definition found!\MessageBreak
253         Maybe you are using a unsupported float version}%
254 }
255 </float & body>

```

6.5 Der floatrow-Hack

Das floatrow-Paket verwendet das Makro `\float@listhead` zum Setzen der Überschriften. Dies wird seit KOMA-Script 3 nicht mehr empfohlen und fliegt demnächst komplett aus der Unterstützung. Stattdessen wird empfohlen, dass Pakete `tocbasic` unterstützen. Der Aufwand dafür ist sehr gering und wird mit vielen neuen Möglichkeiten belohnt.

Dieser Hack rüstet die `tocbasic`-Unterstützung für floatrow nach.

floatrow

```

256 <*package & option>
257 \KOMA@ifkey{floatrow}{@scrhack@floatrow}
258 \KOMAExecuteOptions{floatrow=true}
259 </package & option>
260 <*package & body>
261 \AfterPackage*{floatrow}{%
262     \KOMA@key[.scrhack.sty]{floatrow}{%
263         \PackageWarning{scrhack}{option 'floatrow' ignored}%
264         \FamilyKeyStateProcessed
265     }%
266     \if@scrhack@floatrow\scr@hack@load\@pkgextension{floatrow}\fi
267 }
268 </package & body>

```

`\DeclareNewFloatType` Über die Anweisung `\DeclareNewFloatType` wird eine neue Gleitumgebung definiert. Hier muss die neue Erweiterung aus dem dritten Argument `tocbasic` bekannt gemacht werden.

`\listof` Über die Anweisung `\listof` wird ein Verzeichnis für Gleitumgebungen ausgegeben. Hier muss schlicht die entsprechende Anweisung von `tocbasic` verwendet werden.

`\float@addtolists` Diese Anweisung wird nicht länger benötigt und daher auf die ursprüngliche Definition zurückgesetzt.

```

269 <*floatrow & body>
270 \scr@ifexpected{\DeclareNewFloatType}{%
271   \long\def\DeclareNewFloatType#1#2{\def\FB@captype{#1}%
272     \expandafter\edef\csname ftype@#1\endcsname{\the\c@float@type}%
273     \addtocounter{float@type}{\value{float@type}}%
274     \@namedef{#1name}{#1}\newcounter{#1}%
275     \expandafter\edef\csname fnum@#1\endcsname
276     {\expandafter\noexpand\csname #1name\endcsname\nobreakspace
277     \expandafter\noexpand\csname the#1\endcsname}%
278     \@namedef{the#1}{\arabic{#1}}\flnew@ext{lo#1}\@namedef{fps@#1}{tbp}%
279     \@namedef{l@#1}{\@dottedtocline{1}{1.5em}{2.3em}}%
280     \caption@setkeys[floatrow]{newfloat}{#2}\let\FR@tmp=\relax
281     \xdef\@tempa{\noexpand\flrow@types{\the\flrow@types \FR@tmp{#1}}}%
282     \@tempa}%
283 }{%
284   \scr@ifexpected{\listof}{%
285     \def\listof#1#2{%
286       \@ifundefined{ext@#1}{\flrow@error{Unknown float style ‘#1’}}{%
287         \expandafter\providecommand\csname l@#1\endcsname
288         {\@dottedtocline{1}{1.5em}{2.3em}}%
289         \float@listhead{#2}%
290         \begingroup\setlength{\parskip}{\z@}%
291         \@starttoc{\@nameuse{ext@#1}}%
292         \endgroup}}%
293   }{%
294     \RequirePackage{tocbasic}%
295     \PackageInfo{scrhack}{redefining \string\DeclareNewFloatType}%

```

Eigentlich wäre es besser, wie im `float`-Hack einen Test vorzuschalten, ob die Dateieindung bereits in Gebrauch ist. Aber das würde voraussetzen, dass die Reihenfolge der Anweisungen geändert wird. Dazu stecke ich aber im Code von `floatrow` zu wenig drin. (*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!*)

```

296   \renewcommand\DeclareNewFloatType[2]{\def\FB@captype{#1}%
297     \expandafter\edef\csname ftype@#1\endcsname{\the\c@float@type}%
298     \addtocounter{float@type}{\value{float@type}}%
299     \@namedef{#1name}{#1}\newcounter{#1}%
300     \expandafter\edef\csname fnum@#1\endcsname
301     {\expandafter\noexpand\csname #1name\endcsname\nobreakspace

```

```

302         \expandafter\noexpand\csname the#1\endcsname}%
303         \@namedef{the#1}{\arabic{#1}}\flnew@ext{lo#1}\@namedef{fps@#1}{tbp}%
304         \@namedef{l@#1}{\@dottedtocline{1}{1.5em}{2.3em}}%
305         \caption@setkeys[floatrow]{newfloat}{#2}\let\FR@tmp=\relax
306         \xdef\@tempa{\noexpand\flrow@types{\the\flrow@types \FR@tmp{#1}}}%
307         \@tempa
308         \xdef\@tempa{\noexpand\addtotoclist[float]{\@nameuse{ext@FB@captype}}%
309         \noexpand\setuptoc{\@nameuse{ext@FB@captype}}{chapteratlist}%
310         }%
311         \@tempa
312     }%
313     \PackageInfo{scrhack}{redefining \string\listof}%
314     \renewcommand*\listof[2]{%
315         \@ifundefined{ext@#1}{\flrow@error{Unknown float style ‘#1’}}{%
316             \@ifundefined{l@#1}{\expandafter\let\csname l@#1\endcsname\l@figure
317                 \@ifundefined{l@#1}{%
318                     \@namedef{l@#1}{\@dottedtocline{1}{1.5em}{2.3em}}}%
319                 }{%
320                     \listoftoc[{#2}]{\csname ext@#1\endcsname}%
321                 }%
322             }%
323     \scr@ifexpected{\float@addtolists}{%
324         \long\def\float@addtolists#1{%
325             \def\float@do##1{\addtocontents{##1}{#1}} \the\float@exts}%
326         }{%
327             \PackageInfo{scrhack}{undefining \string\float@addtolists}%
328             \let\float@addtolists\relax
329         }%
330         \PackageWarningNoLine{scrhack}{unkown \string\float@addtolists\space
331             definition found!\MessageBreak
332             Maybe you are using a unsupported floatrow version}%
333     }%
334 }{%
335     \PackageWarningNoLine{scrhack}{unknown \string\listof\space
336         definition found!\MessageBreak
337         Maybe you are using a unsupported floatrow version}%
338 }%
339 }{%
340     \PackageWarningNoLine{scrhack}{unknown \string\DeclareNewFloatType\space
341         definition found!\MessageBreak
342         Maybe you are using a unsupported floatrow version}%
343 }
344 \floatrow & body>

```

6.6 Der listings-Hack

Das listings-Paket verwendet das Makro `\float@listhead` zum Setzen der Überschriften. Dies wird seit KOMA-Script 3 nicht mehr empfohlen und

fliegt demnächst komplett aus der Unterstützung. Stattdessen wird empfohlen, dass Pakete `tocbasic` unterstützen. Der Aufwand dafür ist sehr gering und wird mit vielen neuen Möglichkeiten belohnt.

Dieser Hack rüstet die `tocbasic`-Unterstützung für `listings` nach.

`listings`

```

345 <*package & option>
346 \KOMA@ifkey{listings}{@scrhack@listings}
347 \KOMAExecuteOptions{listings=true}
348 </package & option>
349 <*package & body>
350 \AfterPackage*{listings}{%
351   \KOMA@key[.scrhack.sty]{listings}{%
352     \PackageWarning{scrhack}{option 'listings' ignored}%
353     \FamilyKeyStateProcessed
354   }%
355   \if@scrhack@listings\scr@hack@load\@pkgextension{listings}\fi
356 }
357 </package & body>

```

`\scr@do@hack@listings` Über dieses Macro wird das Verzeichnis der Listings gesetzt. Die gesamte
`\lstlistoflistings` Funktionalität dafür kann `tocbasic` überlassen werden.

`\float@addtolists` Diese Anweisung wird nicht länger benötigt und daher auf die ursprüngliche
Definition zurückgesetzt. Da `listings` ihre Definition mit `\AtBeginDocument` verzögert, muss dies hier ebenfalls geschehen.

```

358 <*listings & body>
359 \newcommand*{\scr@do@hack@listings}{%
360   \RequirePackage{tocbasic}%
361   \addtotoclist[float]{lol}%
362   \setuptoc{lol}{chapteratlist}%
363   \PackageInfo{scrhack}{redefining \string\lstlistoflistings}%
364   \renewcommand*{\lstlistoflistings}{\listoftoc[\lstlistlistingname]{lol}}%
365   \AtBeginDocument{%
366     \scr@ifexpected{\float@addtolists}{%
367       \def\float@addtolists##1{\addtocontents{lol}{##1}}%
368     }{%
369       \PackageInfo{scrhack}{undefining \string\float@addtolists}%
370       \let\float@addtolists\relax
371     }%
372     \scr@ifexpected{\float@addtolists}{%
373       \def\float@addtolists##1{\addtocontents{lol}{##1}}%
374       \orig@float@addtolists{##1}%
375     }{%
376       \PackageInfo{scrhack}{setting \string\float@addtolists\MessageBreak

```

```

377         to \string\orig@float@addtolists}%
378         \let\float@addtolists\orig@float@addtolists
379     }{%
380         \PackageWarningNoLine{scrhack}{unkown \string\float@addtolists\space
381         definition found!\MessageBreak
382         Maybe you are using a unsupported listings version}%
383     }%
384 }%
385 }%
386 \let\scr@do@hack@listings\relax
387 }
388 \scr@ifexpected{\lstlistoflistings}{%
389     \def\lstlistoflistings{\bgroup
390         \let\contentsname\lstlistlistingname
391         \let\lst@temp\@starttoc \def\@starttoc##1{\lst@temp{lol}}%
392         \tableofcontents \egroup}%
393 }{%
394     \scr@do@hack@listings
395 }{%
396     \scr@ifexpected{\lstlistoflistings}{%
397         \def\lstlistoflistings{%
398             \begingroup
399             \@ifundefined{@restonecoltrue}{}{%
400                 \if@twocolumn
401                     \@restonecoltrue\onecolumn
402                 \else
403                     \@restonecolfalse
404                 \fi
405             }%
406             \float@listhead{\lstlistlistingname}%
407             \parskip\z@\parindent\z@\parfillskip \z@ \@plus 1fil%
408             \@starttoc{lol}%
409             \@ifundefined{@restonecoltrue}{}{%
410                 \if@restonecol\twocolumn\fi
411             }%
412             \endgroup
413         }%
414     }{%
415         \scr@do@hack@listings
416     }{%
417         \PackageWarningNoLine{scrhack}{unknown \string\lstlistoflistings\space
418         definition found!\MessageBreak
419         Maybe you are using a unsupported listings version}%
420     }%
421 }
422 </listings & body>

```


6.7 Der setspace-Hack

Das `setspace`-Paket verwendet `\@ptsize` auf ungünstige Art, indem es davon ausgeht, dass es immer eine ganze Zahl enthält. Das ist aber bei KOMA-Script keineswegs zwingend. Außerdem ist der Wert für `11pt` falsch, weil \LaTeX in diesem Fall tatsächlich eine 10,95 pt-Schrift mit einem Zeilenabstand von 13,6 pt einstellt. Damit wäre der korrekte Wert für `\onehalfspacing` beispielsweise:

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

Tatsächlich stellt `setspace` aber einen Wert von 1,213 ein, was einer effektiven Schriftgröße von 11 pt entsprechen würde. Ebenso stellt es den aktuellen Abstand bei `\onehalfspacing` nicht relativ zur aktuellen Schriftgröße ein, sondern zur Grundschriftgröße. Damit erhält man bei

```
\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}
```

einen anderen Abstand als bei

```
\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}
```

obwohl beide Male dieselbe Schriftgröße verwendet wird. Streng genommen müsste also bei jeder Änderung der Schriftgröße der Wert Abstand angepasst werden. So weit geht dieser Hack nicht. Stattdessen wird der Wert abhängig von der tatsächlichen Schriftgröße und dem tatsächlichen Basisabstand beim Aufruf der Anweisungen eingestellt. Das ergibt immerhin in den obigen Beispielen gleiche Ergebnisse.

setspace

```
423 <*package & option>
424 \KOMACheckkey{setspace}{@scrhack@setspace}
425 \KOMAEecuteOptions{setspace=true}
426 </package & option>
427 <*package & body>
428 \AfterPackage*{setspace}{%
429   \KOMACheckkey[.scrhack.sty]{setspace}{%
430     \PackageWarning{scrhack}{option 'setspace' ignored}%
431     \FamilyKeyStateProcessed
432   }%
433   \if@scrhack@setspace\scr@hack@load\@pkgextension{setspace}\fi
434 }
435 </package & body>
```

`\onehalfspacing` Über diese Anweisung wird der eineinhalbzeilige Satz eingestellt. Ein auf drei Nachkommastellen genauer Wert erscheint mir ausreichend genau.

```
436 <*setspace & body>
437 \scr@ifexpected{\onehalfspacing}{%
438   \long\def\onehalfspacing{%
439     \setstretch{1.25}% default
440     \ifcase \@ptsize \relax % 10pt
441       \setstretch {1.25}%
442     \or % 11pt
443       \setstretch {1.213}%
444     \or % 12pt
445       \setstretch {1.241}%
446     \fi
447   }%
448 }{%
449   \renewcommand*\onehalfspacing{%
450     \@tempdima=\dimexpr (\f@size pt)*1500/
451                       (\dimexpr \f@baselineskip\relax)*\p@/1000\relax
452     \expandafter\setstretch\expandafter{\strip@pt\@tempdima}%
453   }%
454 }{%
455   \PackageWarning{scrhack}{unknown \string\onehalfspacing\space
456     definition found!\MessageBreak
457     Maybe you are using a unsupported setpace version}%
458 }
```

`\doublespacing` Über diese Anweisung wird der zweizeilige Satz eingestellt. Ein auf drei Nachkommastellen genauer Wert erscheint mir ausreichend genau.

```
459 \scr@ifexpected{\doublespacing}{%
460   \long\def\doublespacing{%
461     \setstretch{1.667}% default
```

```

462 \ifcase \@ptsize \relax % 10pt
463 \setstretch {1.667}%
464 \or % 11pt
465 \setstretch {1.618}%
466 \or % 12pt
467 \setstretch {1.655}%
468 \fi
469 }%
470 }{%
471 \renewcommand*{\doublespacing}{%
472 \@tempdima=\dimexpr (\f@size pt)*2000/
473 (\dimexpr \f@baselineskip\relax)*\p@/1000\relax
474 \expandafter\setstretch\expandafter{\strip@pt\@tempdima}%
475 }%
476 }{%
477 \PackageWarning{scrhack}{unknown \string\doublespacing\space
478 definition found!\MessageBreak
479 Maybe you are using a unsupported setpace version}%
480 }
481 </setspace & body>

```

6.8 Optionen ausführen

Zum Schluss noch die Optionen ausführen. Im Paket wird diese Anweisung allerdings vor den Anweisungen der Hacks und den Anweisungen aus dem Abschnitt »Verwendete Anweisungen« stehen.

```

482 <*package & option>
483 \KOMAProcessOptions\relax
484 </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.

Symbols		F	
<code>\@schapter</code>	<u>50</u>	<code>float (Option)</code>	<u>154</u>
<code>\@spart</code>	<u>50</u>	<code>\float@addtolists</code> ..	<u>167</u> , <u>269</u> , <u>358</u>
<code>\@ssect</code>	<u>50</u>	<code>floatrow (Option)</code>	<u>256</u>
D		H	
<code>\DeclareNewFloatType</code>	<u>269</u>	<code>hyperref (Option)</code>	<u>28</u>
<code>\doublespacing</code>	<u>459</u>		

L		
listings (Option)	345	listings
\listof	167, 269	setspace
\lstlistoflistings	358	
N		S
\newfloat	167	\scr@do@hack@listings
O		\scr@hack@load
\onehalfspacing	436	\scr@ifexpected
Optionen:		secnumdepth (Zähler)
floatrow	256	setspace (Option)
float	154	
hyperref	28	Z
		Zähler:
		secnumdepth

Change History

v3.03		floatrow: Signalisierung mit
secnumdepth: erste Version des		\FamilyKeyStateProcessed
Pakets	5 12
v3.04b		hyperref: Signalisierung mit
secnumdepth: Die Reihenfolge		\FamilyKeyStateProcessed
von Anweisungen und Op-	 6
tionen grundlegend geändert,		v3.17
um das Paket scrhack unab-		\doublespacing: Neu
hängiger von der Reihenfolge		setspace: Neu
beim Laden von Paketen zu		listings: Defaulteinstellung
machen.	5	mit \KOMAEecuteOptions 15
hyperref: hyperref-Hack wird		float: Defaulteinstellung mit
früher geladen	6	\FOMAEecuteOptions 9
v3.12		floatrow: Defaulteinstellung
listings: Signalisierung mit		mit \KOMAEecuteOptions 12
\FamilyKeyStateProcessed		\onehalfspacing: Neu
..... 15		hyperref: Defaulteinstellung
float: Signalisierung mit		mit \KOMAEecuteOptions . 6
\FamilyKeyStateProcessed		Neuere hyperref-Version deak-
..... 9		tiviert ggf. Option hyperref 6