

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	3
5	Implementation of <code>scrhack</code>	4
5.1	Optionen	4
5.2	Verwendete Anweisungen	5
5.3	Der hyperref-Hack	5
5.4	Der float-Hack	9
5.5	Der floatrow-Hack	11
5.6	Der listings-Hack	14
5.7	Optionen ausführen	16

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

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

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 compatiblity 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 `\listofflistings`, 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 Implementation of `scrhack`

5.1 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.

5.2 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 L^AT_EX 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 }
```

5.3 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 \@scrhack@hyperreftrue
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   }{%
39     \AtEndOfPackage{%
40       \KOMA@key[.scrhack.sty]{hyperref}{%
41         \PackageWarning{scrhack}{option 'hyperref=#1' ignored}%
42         \FamilyKeyStateProcessed
43       }%
44       \if@scrhack@hyperref\scr@hack@load\@pkgextension{hyperref}\fi
45     }%
46   }%
47 }
48 </package & body>
```

`\@schapter` Eigentlich wird hier gar nicht `hyperref.sty` verändert, sondern diverse
`\@spart` Treiberdateien. Sobald das Paket `hyperref` geladen ist, ist auch die passende
`\@ssect` Treiberdatei geladen und außerdem sind alle Treiberdateien, die entsprechende
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 sichergestellt werden, dass die
umzudefinierenden Macros derzeit den erwarteten Inhalt haben.

```
49 <*hyperref & body>
50 \scr@ifexpected\@schapter{%
51   \def\@schapter#1{%
52     \H@old@schapter{#1}%
53     \begingroup
```

```

54     \let\@mkboth\@gobbletwo
55     \Hy@GlobalStepCount\Hy@linkcounter
56     \xdef\@currentHref{\Hy@chapapp*.\the\Hy@linkcounter}%
57     \Hy@raisedlink{%
58         \hyper@anchorstart{\@currentHref}\hyper@anchorend
59     }%
60 \endgroup
61 }%
62 }{%
63 \PackageInfo{scrhack}{redefining \string\@schapter}%
64 \def\@schapter#1{%
65     \begingroup
66     \let\@mkboth\@gobbletwo
67     \Hy@GlobalStepCount\Hy@linkcounter
68     \xdef\@currentHref{\Hy@chapapp*.\the\Hy@linkcounter}%
69     \Hy@raisedlink{%
70         \hyper@anchorstart{\@currentHref}\hyper@anchorend
71     }%
72 \endgroup
73 \H@old@schapter{#1}%
74 }%
75 }{%
76 \scr@ifexpected\@schapter{%
77     \def\@schapter#1{%
78         \begingroup
79         \let\@mkboth\@gobbletwo
80         \Hy@GlobalStepCount\Hy@linkcounter
81         \xdef\@currentHref{\Hy@chapapp*.\the\Hy@linkcounter}%
82         \Hy@raisedlink{%
83             \hyper@anchorstart{\@currentHref}\hyper@anchorend
84         }%
85     \endgroup
86     \H@old@schapter{#1}%
87 }%
88 }{}{%
89 \PackageWarningNoLine{scrhack}{unknown \string\@schapter\space
90     definition found!\MessageBreak
91     Maybe you are using a unsupported hyperref version}%
92 }%
93 }
94
95 \scr@ifexpected\@spart{%
96 \def\@spart#1{%
97     \H@old@spart{#1}%
98     \Hy@GlobalStepCount\Hy@linkcounter
99     \xdef\@currentHref{part*.\the\Hy@linkcounter}%
100 \Hy@raisedlink{%
101     \hyper@anchorstart{\@currentHref}\hyper@anchorend
102 }%

```

```

103 }%
104 }{%
105 \PackageInfo{scrhack}{redefining \string\@spart}%
106 \def\@spart#1{%
107   \Hy@GlobalStepCount\Hy@linkcounter
108   \xdef\@currentHref{part*.\the\Hy@linkcounter}%
109   \Hy@raisedlink{%
110     \hyper@anchorstart{\@currentHref}\hyper@anchorend
111   }%
112   \H@old@spart{#1}%
113 }%
114 }{%
115 \scr@ifexpected\@spart{%
116   \def\@spart#1{%
117     \Hy@GlobalStepCount\Hy@linkcounter
118     \xdef\@currentHref{part*.\the\Hy@linkcounter}%
119     \Hy@raisedlink{%
120       \hyper@anchorstart{\@currentHref}\hyper@anchorend
121     }%
122     \H@old@spart{#1}%
123   }%
124 }{}{%
125   \PackageWarningNoLine{scrhack}{unknown \string\@spart\space
126     definition found!\MessageBreak
127     Maybe you are using a unsupported hyperref version}%
128 }%
129 }
130
131 \scr@ifexpected\@ssect{%
132   \def\@ssect#1#2#3#4#5{%
133     \H@old@ssect{#1}{#2}{#3}{#4}{#5}%
134     \phantomsection
135   }%
136 }{%
137 \PackageInfo{scrhack}{redefining \string\@ssect}%
138 \def\@ssect#1#2#3#4#5{%
139   \H@old@ssect{#1}{#2}{#3}{#4}{\phantomsection\ignorespaces#5}%
140 }%
141 }{%
142 \scr@ifexpected\@ssect{%
143   \def\@ssect#1#2#3#4#5{%
144     \H@old@ssect{#1}{#2}{#3}{#4}{\phantomsection\ignorespaces#5}%
145   }%
146 }{}{%
147   \PackageWarningNoLine{scrhack}{unknown \string\@ssect\space
148     definition found!\MessageBreak
149     Maybe you are using a unsupported hyperref version}%
150 }%
151 }

```


152 \langle /hyperref & body \rangle

5.4 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`

```
153  $\langle$ *package & option $\rangle$ 
154 \KOMA@ifkey{float}{@scrhack@float}%
155 \@scrhack@floattrue
156  $\langle$ /package & option $\rangle$ 
157  $\langle$ *package & body $\rangle$ 
158 \AfterPackage*{float}{%
159   \KOMA@key[.scrhack.sty]{float}{%
160     \PackageWarning{scrhack}{option 'float' ignored}%
161     \FamilyKeyStateProcessed
162   }%
163   \if@scrhack@float\scr@hack@load\@pkgextension{float}\fi
164 }
165  $\langle$ /package & body $\rangle$ 
```

`\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.

```
166  $\langle$ *float & body $\rangle$ 
167 \scr@ifexpected{\newfloat}{%
168   \long\def\newfloat#1#2#3{\@namedef{ext@#1}{#3}
169     \let\float@do=\relax
170     \xdef\@tempa{\noexpand\float@exts{\the\float@exts \float@do{#3}}}%
171     \@tempa
172     \floatplacement{#1}{#2}%
173   }%
174 }
```

```

173 \ifundefined{fname@#1}{\floatname{#1}{#1}}{}
174 \expandafter\edef\csname ftype@#1\endcsname{\value{float@type}}%
175 \addtocounter{float@type}{\value{float@type}}
176 \restylefloat{#1}%
177 \expandafter\edef\csname fnum@#1\endcsname%
178 {\expandafter\noexpand\csname fname@#1\endcsname{}
179 \expandafter\noexpand\csname the#1\endcsname}
180 \@ifnextchar[%]
181 {\float@newx{#1}}%
182 {\@ifundefined{c@#1}{\newcounter{#1}\@namedef{the#1}{\arabic{#1}}}%
183 {}}%
184 }{%
185 \scr@ifexpected{\listof}{%
186 \def\listof#1#2{%
187 \ifundefined{ext@#1}{\float@error{#1}}{%
188 \@namedef{l@#1}{\@dottedtocline{1}{1.5em}{2.3em}}%
189 \float@listhead{#2}%
190 \begingroup\setlength{\parskip}{\z@}%
191 \@starttoc{\@nameuse{ext@#1}}%
192 \endgroup}}%
193 }{%
194 \RequirePackage{tocbasic}%
195 \PackageInfo{scrhack}{redefining \string\newfloat}%
196 \renewcommand\newfloat[3]{%
197 \ifattoclist{#3}{%
198 \PackageError{scrhack}{extension ‘#3’ already in use}{%
199 Each extension may be used only once.\MessageBreak
200 You, the class, or another package already uses extension
201 ‘#3’.\MessageBreak
202 \string\newfloat\space command will be ignored!}%
203 }{%
204 \addtotoclist[float]{#3}%
205 \setuptoc{#3}{chapteratlist}%
206 \@namedef{ext@#1}{#3}%
207 \let\float@do=\relax
208 \xdef\@tempa{\noexpand\float@exts{\the\float@exts \float@do{#3}}}%
209 \@tempa
210 \floatplacement{#1}{#2}%
211 \@ifundefined{fname@#1}{\floatname{#1}{#1}}{}%
212 \expandafter\edef\csname ftype@#1\endcsname{\value{float@type}}%
213 \addtocounter{float@type}{\value{float@type}}
214 \restylefloat{#1}%
215 \expandafter\edef\csname fnum@#1\endcsname%
216 {\expandafter\noexpand\csname fname@#1\endcsname{}
217 \expandafter\noexpand\csname the#1\endcsname}%
218 \@ifnextchar[%]
219 {\float@newx{#1}}%
220 {\@ifundefined{c@#1}{\newcounter{#1}\@namedef{the#1}{\arabic{#1}}}%
221 {}}%

```

```

222 }%
223 \PackageInfo{scrhack}{redefining \string\listof}%
224 \renewcommand*\listof[2]{%
225   \@ifundefined{ext@#1}{\float@error{#1}}{%
226     \@ifundefined{l@#1}{\expandafter\let\csname l@#1\endcsname\l@figure
227       \@ifundefined{l@#1}{%
228         \@namedef{l@#1}{\@dottedtocline{1}{1.5em}{2.3em}}{}}{%
229       }{%
230       \listoftoc[{#2}]{\csname ext@#1\endcsname}%
231     }%
232   }%
233 \scr@ifexpected{\float@addtolists}{%
234   \long\def\float@addtolists#1{%
235     \def\float@do##1{\addtocontents{##1}{#1}} \the\float@exts}%
236 }{%
237   \PackageInfo{scrhack}{undefining \string\float@addtolists}%
238   \let\float@addtolists\relax
239 }{%
240   \PackageWarningNoLine{scrhack}{unkown \string\float@addtolists\space
241     definition found!\MessageBreak
242     Maybe you are using a unsupported float version}%
243 }%
244 }{%
245   \PackageWarningNoLine{scrhack}{unknown \string\listof\space
246     definition found!\MessageBreak
247     Maybe you are using a unsupported float version}%
248 }%
249 }{%
250   \PackageWarningNoLine{scrhack}{unknown \string\newfloat\space
251     definition found!\MessageBreak
252     Maybe you are using a unsupported float version}%
253 }
254 </float & body>

```

5.5 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

```

255 <*package & option>

```

```

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

```

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

```

```

292 }{%
293   \RequirePackage{tocbasic}%
294   \PackageInfo{scrhack}{redefining \string\DeclareNewFloatType}%

```

Eigentlich wäre es besser, wie im float-Hack einen Test vorzuschalten, ob die Dateiendung 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!*)

```

295   \renewcommand\DeclareNewFloatType[2]{\def\FB@capttype{#1}%
296     \expandafter\edef\csname ftype@#1\endcsname{\the\c@float@type}%
297     \addtocounter{float@type}{\value{float@type}}%
298     \@namedef{#1name}{#1}\newcounter{#1}%
299     \expandafter\edef\csname fnum@#1\endcsname
300     {\expandafter\noexpand\csname #1name\endcsname\nobreakspace
301     \expandafter\noexpand\csname the#1\endcsname}%
302     \@namedef{the#1}{\arabic{#1}}\flnew@ext{lo#1}\@namedef{fps@#1}{tbp}%
303     \@namedef{l@#1}{\@dottedtocline{1}{1.5em}{2.3em}}%
304     \caption@setkeys[floatrow]{newfloat}{#2}\let\FR@tmp=\relax
305     \xdef\@tempa{\noexpand\flrow@types{\the\flrow@types \FR@tmp{#1}}}%
306     \@tempa
307     \xdef\@tempa{\noexpand\addtotoclist[float]{\@nameuse{ext@FB@capttype}}%
308     \noexpand\setuptoc{\@nameuse{ext@FB@capttype}}{chapteratlist}%
309     }%
310     \@tempa
311   }%
312   \PackageInfo{scrhack}{redefining \string\listof}%
313   \renewcommand*\listof[2]{%
314     \@ifundefined{ext@#1}{\flrow@error{Unknown float style '#1'}}{%
315       \@ifundefined{l@#1}{\expandafter\let\csname l@#1\endcsname\l@figure
316       \@ifundefined{l@#1}{%
317         \@namedef{l@#1}{\@dottedtocline{1}{1.5em}{2.3em}}}{%
318       }}%
319     \listoftoc[{#2}]{\csname ext@#1\endcsname}%
320   }%
321 }%
322 \scr@ifexpected{\float@addtolists}{%
323   \long\def\float@addtolists#1{%
324     \def\float@do##1{\addtocontents{##1}{#1}} \the\float@exts}%
325 }{%
326   \PackageInfo{scrhack}{undefining \string\float@addtolists}%
327   \let\float@addtolists\relax
328 }{%
329   \PackageWarningNoLine{scrhack}{unkown \string\float@addtolists\space
330   definition found!\MessageBreak
331   Maybe you are using a unsupported floatrow version}%
332 }%

```

```

333 }{%
334   \PackageWarningNoLine{scrhack}{unknown \string\listof\space
335     definition found!\MessageBreak
336     Maybe you are using a unsupported floatrow version}%
337 }%
338 }{%
339   \PackageWarningNoLine{scrhack}{unknown \string\DeclareNewFloatType\space
340     definition found!\MessageBreak
341     Maybe you are using a unsupported floatrow version}%
342 }
343 </floatrow & body>

```

5.6 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`

```

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

```

357 <*listings & body>

```

```

358 \newcommand*{\scr@do@hack@listings}{%
359   \RequirePackage{tocbasic}%
360   \addtotoclist[float]{lol}%
361   \setuptoc{lol}{chapteratlist}%
362   \PackageInfo{scrhack}{redefining \string\lstlistoflistings}%
363   \renewcommand*{\lstlistoflistings}{\listoftoc[\lstlistlistingname]{lol}}%
364   \AtBeginDocument{%
365     \scr@ifexpected{\float@addtolists}{%
366       \def\float@addtolists##1{\addtocontents{lol}{##1}}%
367     }{%
368       \PackageInfo{scrhack}{undefining \string\float@addtolists}%
369       \let\float@addtolists\relax
370     }{%
371       \scr@ifexpected{\float@addtolists}{%
372         \def\float@addtolists##1{\addtocontents{lol}{##1}}%
373         \orig@float@addtolists{##1}}%
374       {%
375         \PackageInfo{scrhack}{setting \string\float@addtolists\MessageBreak
376           to \string\orig@float@addtolists}%
377         \let\float@addtolists\orig@float@addtolists
378       }{%
379         \PackageWarningNoLine{scrhack}{unkown \string\float@addtolists\space
380           definition found!\MessageBreak
381           Maybe you are using a unsupported listings version}%
382       }%
383     }%
384   }%
385   \let\scr@do@hack@listings\relax
386 }
387 \scr@ifexpected{\lstlistoflistings}{%
388   \def\lstlistoflistings{\bgroup
389     \let\contentsname\lstlistlistingname
390     \let\lst@temp\@starttoc \def\@starttoc##1{\lst@temp{lol}}%
391     \tableofcontents \egroup}%
392 }{%
393   \scr@do@hack@listings
394 }{%
395   \scr@ifexpected{\lstlistoflistings}{%
396     \def\lstlistoflistings{%
397       \begingroup
398       \ifundefined{@restonecoltrue}{%
399         \if@twocolumn
400           \@restonecoltrue\onecolumn
401         \else
402           \@restonecolfalse
403         \fi
404       }%
405       \float@listhead{\lstlistlistingname}%
406       \parskip\z@\parindent\z@\parfillskip \z@ \@plus 1fil%

```

```

407      \@starttoc{lol}%
408      \@ifundefined{@restonecoltrue}{-}{%
409          \if@restonecol\twocolumn\fi
410      }%
411      \endgroup
412  }%
413 }{%
414     \scr@do@hack@listings
415 }{%
416     \PackageWarningNoLine{scrhack}{unknown \string\lstlistoflistings\space
417         definition found!\MessageBreak
418         Maybe you are using a unsupported listings version}%
419 }%
420 }
421 </listings & body>

```

5.7 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.

```

422 <*package & option>
423 \KOMAProcessOptions\relax
424 </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		L	
\@schapter	<u>49</u>	listings (Option)	<u>344</u>
\@spart	<u>49</u>	\listof	<u>166</u> , <u>268</u>
\@ssect	<u>49</u>	\lstlistoflistings	<u>357</u>
D		N	
\DeclareNewFloatType	<u>268</u>	\newfloat	<u>166</u>
F		O	
float (Option)	<u>153</u>	Optionen:	
\float@addtolists ..	<u>166</u> , <u>268</u> , <u>357</u>	floatrow	<u>255</u>
floatrow (Option)	<u>255</u>	float	<u>153</u>
H		hyperref	<u>28</u>
hyperref (Option)	<u>28</u>	listings	<u>344</u>

S	secnumdepth (Zähler)	1
<code>\scr@do@hack@listings</code>		357
<code>\scr@hack@load</code>		12
<code>\scr@ifexpected</code>		1
	Z	
	Zähler:	
	secnumdepth	1

Change History

v3.03	v3.12
secnumdepth: erste Version des Pakets	listings: Signalisierung mit <code>\FamilyKeyStateProcessed</code>
4	14
v3.04b	float: Signalisierung mit <code>\FamilyKeyStateProcessed</code>
secnumdepth: Die Reihenfolge von Anweisungen und Optionen grundlegend geändert, um das Paket <code>scrhack</code> unabhängiger von der Reihenfolge beim Laden von Paketen zu machen.	9
4	floatrow: Signalisierung mit <code>\FamilyKeyStateProcessed</code>
hyperref: <code>hyperref</code> -Hack wird früher geladen	11
6	hyperref: Signalisierung mit <code>\FamilyKeyStateProcessed</code>
	6