

KOMA-Script File `scrhack.dtx` *

Markus Kohm[†]

2011/09/28

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 v3.10 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 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 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 \KOMACheck{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       \KOMACheck[.scrhack.sty]{hyperref}{%
41         \PackageWarning{scrhack}{option 'hyperref=#1' ignored}%
42       }%
43     \if@scrhack@hyperref\scrhack@load\@pkgextension{hyperref}\fi
44   }%
45 }%
46 }
47 </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 `\scrhack@load` getestet wurde). Es muss also nur noch sicher-
gestellt werden, dass die umzudefinierenden Macros derzeit den erwarteten
Inhalt haben.

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

```

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

```

```

101     }%
102 }%
103 }{%
104   \PackageInfo{scrhack}{redefining \string\@spart}%
105   \def\@spart#1{%
106     \Hy@GlobalStepCount\Hy@linkcounter
107     \xdef\@currentHref{part*.\the\Hy@linkcounter}%
108     \Hy@raisedlink{%
109       \hyper@anchorstart{\@currentHref}\hyper@anchorend
110     }%
111     \H@old@spart{#1}%
112   }%
113 }{%
114   \scr@ifexpected\@spart{%
115     \def\@spart#1{%
116       \Hy@GlobalStepCount\Hy@linkcounter
117       \xdef\@currentHref{part*.\the\Hy@linkcounter}%
118       \Hy@raisedlink{%
119         \hyper@anchorstart{\@currentHref}\hyper@anchorend
120       }%
121       \H@old@spart{#1}%
122     }%
123   }{}{%
124     \PackageWarningNoLine{scrhack}{unknown \string\@spart\space
125       definition found!\MessageBreak
126       Maybe you are using a unsupported hyperref version}%
127   }%
128 }
129
130 \scr@ifexpected\@ssect{%
131   \def\@ssect#1#2#3#4#5{%
132     \H@old@ssect{#1}{#2}{#3}{#4}{#5}%
133     \phantomsection
134   }%
135 }{%
136   \PackageInfo{scrhack}{redefining \string\@ssect}%
137   \def\@ssect#1#2#3#4#5{%
138     \H@old@ssect{#1}{#2}{#3}{#4}{\phantomsection\ignorespaces#5}%
139   }%
140 }{%
141   \scr@ifexpected\@ssect{%
142     \def\@ssect#1#2#3#4#5{%
143       \H@old@ssect{#1}{#2}{#3}{#4}{\phantomsection\ignorespaces#5}%
144     }%
145   }{}{%
146     \PackageWarningNoLine{scrhack}{unknown \string\@ssect\space
147       definition found!\MessageBreak
148       Maybe you are using a unsupported hyperref version}%
149   }%

```



```

150 }
151 </hyperref & body>

```

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`

```

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

```

164 <*float & body>
165 \scr@ifexpected{\newfloat}{%
166   \long\def\newfloat#1#2#3{\@namedef{ext@#1}{#3}
167     \let\float@do=\relax
168     \xdef\@tempa{\noexpand\float@exts{\the\float@exts \float@do{#3}}}%
169     \@tempa
170     \floatplacement{#1}{#2}%

```

```

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

```

```

220 }%
221 \PackageInfo{scrhack}{redefining \string\listof}%
222 \renewcommand*\listof[2]{%
223   \@ifundefined{ext@#1}{\float@error{#1}}{%
224     \@ifundefined{l@#1}{\expandafter\let\csname l@#1\endcsname\l@figure
225       \@ifundefined{l@#1}{%
226         \@namedef{l@#1}{\@dottedtocline{1}{1.5em}{2.3em}}{}}{%
227       }{%
228       \listoftoc[{#2}]{\csname ext@#1\endcsname}%
229     }%
230   }%
231   \scr@ifexpected{\float@addtolists}{%
232     \long\def\float@addtolists#1{%
233       \def\float@do##1{\addtocontents{##1}{#1}} \the\float@exts}%
234   }{%
235     \PackageInfo{scrhack}{undefining \string\float@addtolists}%
236     \let\float@addtolists\relax
237   }{%
238     \PackageWarningNoLine{scrhack}{unkown \string\float@addtolists\space
239       definition found!\MessageBreak
240       Maybe you are using a unsupported float version}%
241   }%
242 }{%
243   \PackageWarningNoLine{scrhack}{unknown \string\listof\space
244     definition found!\MessageBreak
245     Maybe you are using a unsupported float version}%
246 }%
247 }{%
248   \PackageWarningNoLine{scrhack}{unknown \string\newfloat\space
249     definition found!\MessageBreak
250     Maybe you are using a unsupported float version}%
251 }
252 </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

```

253 <*package & option>

```

```

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

```

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

```

```

290 \RequirePackage{tocbasic}%
291 \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!*)

```

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

```

```

331 \PackageWarningNoLine{scrhack}{unknown \string\listof\space
332 definition found!\MessageBreak
333 Maybe you are using a unsupported floatrow version}%
334 }%
335 }{%
336 \PackageWarningNoLine{scrhack}{unknown \string\DeclareNewFloatType\space
337 definition found!\MessageBreak
338 Maybe you are using a unsupported floatrow version}%
339 }
340 </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`

```

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

```

353 <*listings & body>
354 \newcommand*{\scr@do@hack@listings}{%
355 \RequirePackage{tocbasic}%

```

```

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

```

```

405         \if@restonecol\twocolumn\fi
406     }%
407 \endgroup
408 }%
409 }{%
410     \scr@do@hack@listings
411 }{%
412     \PackageWarningNoLine{scrhack}{unknown \string\lstlistoflistings\space
413         definition found!\MessageBreak
414         Maybe you are using a unsupported listings version}%
415 }%
416 }
417 </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.

```

418 <*package & option>
419 \KOMAProcessOptions\relax
420 </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>48</u>	listings (Option)	<u>341</u>
\@spart	<u>48</u>	\listof	<u>164</u> , <u>265</u>
\@ssect	<u>48</u>	\lstlistoflistings	<u>353</u>
D		N	
\DeclareNewFloatType	<u>265</u>	\newfloat	<u>164</u>
F		O	
float (Option)	<u>152</u>	Optionen:	
\float@addtolists ..	<u>164</u> , <u>265</u> , <u>353</u>	floatrow	<u>253</u>
floatrow (Option)	<u>253</u>	float	<u>152</u>
		hyperref	<u>28</u>
		listings	<u>341</u>
H		S	
hyperref (Option)	<u>28</u>	\scr@do@hack@listings	<u>353</u>

\scr@hack@load	<u>12</u>	Z
\scr@ifexpected	<u>1</u>	Zähler:
secnumdepth (Zähler)	<u>1</u>	secnumdepth
		<u>1</u>

Change History

v3.03		um das Paket scrhack unab-
secnumdepth: erste Version des		hängiger von der Reihenfolge
Pakets	4	beim Laden von Paketen zu
v3.04b		machen.
secnumdepth: Die Reihenfolge		4
von Anweisungen und Op-		
tionen grundlegend geändert,		hyperref: hyperref-Hack wird
		früher geladen
		6