

# KOMA-Script File `scrhack.dtx` \*

Markus Kohm<sup>†</sup>

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

<b>1</b>	<b>The hyperref hack</b>	<b>2</b>
<b>2</b>	<b>The float hack</b>	<b>2</b>
<b>3</b>	<b>The floatrow hack</b>	<b>3</b>
<b>4</b>	<b>The listings hack</b>	<b>4</b>
<b>5</b>	<b>The setspace hack</b>	<b>4</b>
<b>6</b>	<b>The lscape hack</b>	<b>5</b>

---

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

<sup>†</sup>[mailto:komascript\(at\)gmx.info](mailto:komascript(at)gmx.info)

<b>7</b>	<b>Implementation of scrhack</b>	<b>5</b>
7.1	Optionen . . . . .	5
7.2	Verwendete Anweisungen . . . . .	6
7.3	Der hyperref-Hack . . . . .	7
7.4	Der float-Hack . . . . .	10
7.5	Der floatrow-Hack . . . . .	13
7.6	Der listings-Hack . . . . .	15
7.7	Der setspace-Hack . . . . .	17
7.8	Der lscape-Hack . . . . .	20
7.9	Optionen ausführen . . . . .	21

## 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 The `lscape` hack

Package `lscape` defines an environment `landscape` to set the page contents but not head and foot landscape. Inside this environment it changes `\textheight` to the value of `\textwidth`, but it does not change `\textwidth` to the former value of `\textheight`. This is inconsistent. As far as I know, `\textwidth` is unchanged because setting it to `\textheight` could blame other packages or user commands. But changing `\textheight` could also blame other packages or user commands and indeed it breaks, e. g., `showframe` and `sclayer`. So best would be, not to change `\textheight`, too. `scrhack` uses package `xpatch` to modify the environment start macro `\landscape` appropriately.

You can switch of the `lscape` hack loading package `scrhack` with option `lscape=false`. You can also change option `lscape` afterwards. If the option is `false` while loading `lscape`, `scrhack` will not patch `\landscape` and later changes of the option have no effect. But if the option is `true` while loading `lscape` or if `scrhack` is loaded after `lscape` without option `lscape=false`, every later change of the option using `\KOMAoption` or `\KOMAOptions` will have the expected effect.

## 7 Implementation of `scrhack`

### 7.1 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.

## 7.2 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<sup>A</sup>T<sub>E</sub>X 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 }
```

### 7.3 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 \KOMACheckkey{hyperref}{@scrhack@hyperref}%
30 \KOMAExecuteOptions{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 \BeforePackage{hyperref}{%
34   \scr@ifundefinedorrelax{hy@insteadofrefstepcounter}{%
35     \PackageInfo{scrhack}{hyperref hack deactivated because of\MessageBreak
36       detection of KOMA-Script class, that doesn't\MessageBreak
37       need that hack,}%
38     \KOMAExecuteOptions[.scrhack.sty]{hyperref=false}%
39   }%
40 }
41 \AfterPackage*{hyperref}{%
42   \if@scrhack@hyperref
43     \@ifpackagelater{hyperref}{2009/11/24}{%
44       \PackageInfo{scrhack}{hyperref hack deactivated because of\MessageBreak
45         detection of hyperref version, that doesn't\MessageBreak
46         need that hack,}%
47       \KOMAExecuteOptions[.scrhack.sty]{hyperref=false}%
48     }{%
49       \AtEndOfPackage{%
50         \KOMACheckkey[.scrhack.sty]{hyperref}{%
51           \PackageWarning{scrhack}{option 'hyperref=#1' ignored}%
52           \FamilyKeyStateProcessed
53         }%
54       }
55     }
56   }%
57 }
```

```

54         \if@scrhack@hyperref\scr@hack@load\@pkgextension{hyperref}\fi
55     }%
56 }%
57 \fi
58 }
59 </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.

```

60 <*hyperref & body>
61 \scr@ifexpected\@schapter{%
62     \def\@schapter#1{%
63         \H@old@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     }%
73 }{%
74     \PackageInfo{scrhack}{redefining \string\@schapter}%
75     \def\@schapter#1{%
76         \begingroup
77             \let\@mkboth\@gobbletwo
78             \Hy@GlobalStepCount\Hy@linkcounter
79             \xdef\@currentHref{\Hy@chapapp*.\the\Hy@linkcounter}%
80             \Hy@raisedlink{%
81                 \hyper@anchorstart{\@currentHref}\hyper@anchorend
82             }%
83         \endgroup
84         \H@old@schapter{#1}%
85     }%
86 }{%
87     \scr@ifexpected\@schapter{%
88         \def\@schapter#1{%
89             \begingroup
90                 \let\@mkboth\@gobbletwo
91                 \Hy@GlobalStepCount\Hy@linkcounter

```



```

92         \xdef\@currentHref{\Hy@chapapp*.\the\Hy@linkcounter}%
93         \Hy@raisedlink{%
94             \hyper@anchorstart{\@currentHref}\hyper@anchorend
95         }%
96     \endgroup
97     \H@old@schapter{#1}%
98 }%
99 }{}{%
100     \PackageWarningNoLine{scrhack}{unknown \string\@schapter\space
101         definition found!\MessageBreak
102         Maybe you are using a unsupported hyperref version}%
103 }%
104 }
105
106 \scr@ifexpected\@spart{%
107     \def\@spart#1{%
108         \H@old@spart{#1}%
109         \Hy@GlobalStepCount\Hy@linkcounter
110         \xdef\@currentHref{part*.\the\Hy@linkcounter}%
111         \Hy@raisedlink{%
112             \hyper@anchorstart{\@currentHref}\hyper@anchorend
113         }%
114     }%
115 }{%
116     \PackageInfo{scrhack}{redefining \string\@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     \scr@ifexpected\@spart{%
127         \def\@spart#1{%
128             \Hy@GlobalStepCount\Hy@linkcounter
129             \xdef\@currentHref{part*.\the\Hy@linkcounter}%
130             \Hy@raisedlink{%
131                 \hyper@anchorstart{\@currentHref}\hyper@anchorend
132             }%
133             \H@old@spart{#1}%
134         }%
135     }{}{%
136         \PackageWarningNoLine{scrhack}{unknown \string\@spart\space
137             definition found!\MessageBreak
138             Maybe you are using a unsupported hyperref version}%
139     }%
140 }

```

```

141
142 \scr@ifexpected\@ssect{%
143   \def\@ssect#1#2#3#4#5{%
144     \H@old@ssect{#1}{#2}{#3}{#4}{#5}%
145     \phantomsection
146   }%
147 }{%
148   \PackageInfo{scrhack}{redefining \string\@ssect}%
149   \def\@ssect#1#2#3#4#5{%
150     \H@old@ssect{#1}{#2}{#3}{#4}{\phantomsection\ignorespaces#5}%
151   }%
152 }{%
153   \scr@ifexpected\@ssect{%
154     \def\@ssect#1#2#3#4#5{%
155       \H@old@ssect{#1}{#2}{#3}{#4}{\phantomsection\ignorespaces#5}%
156     }%
157   }{}{%
158     \PackageWarningNoLine{scrhack}{unknown \string\@ssect\space
159       definition found!\MessageBreak
160       Maybe you are using a unsupported hyperref version}%
161   }%
162 }
163 </hyperref & body>

```

## 7.4 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

```

164 <*package & option>
165 \KOMA@ifkey{float}{@scrhack@float}%
166 \KOMAExecuteOptions{float=true}%
167 </package & option>
168 <*package & body>
169 \AfterPackage*{float}{%
170   \KOMA@key[.scrhack.sty]{float}{%
171     \PackageWarning{scrhack}{option 'float' ignored}%
172     \FamilyKeyStateProcessed
173   }%
174   \if@scrhack@float\scr@hack@load\@pkgextension{float}\fi

```

```

175 }
176 </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.

```

177 <*float & body>
178 \scr@ifexpected{\newfloat}{%
179   \long\def\newfloat#1#2#3{\@namedef{ext@#1}{#3}
180     \let\float@do=\relax
181     \xdef\@tempa{\noexpand\float@exts{\the\float@exts \float@do{#3}}}%
182     \@tempa
183     \floatplacement{#1}{#2}%
184     \@ifundefined{fname@#1}{\floatname{#1}{#1}}{}
185     \expandafter\edef\csname ftype@#1\endcsname{\value{float@type}}%
186     \addtocounter{float@type}{\value{float@type}}
187     \restylefloat{#1}%
188     \expandafter\edef\csname fnum@#1\endcsname%
189     {\expandafter\noexpand\csname fname@#1\endcsname{}
190      \expandafter\noexpand\csname the#1\endcsname}
191     \@ifnextchar[%]
192     {\float@newx{#1}}%
193     {\@ifundefined{c@#1}{\newcounter{#1}\@namedef{the#1}{\arabic{#1}}}%
194      {}}}%
195 }{%
196   \scr@ifexpected{\listof}{%
197     \def\listof#1#2{%
198       \@ifundefined{ext@#1}{\float@error{#1}}{%
199         \@namedef{l@#1}{\@dottedtocline{1}{1.5em}{2.3em}}%
200         \float@listhead{#2}%
201         \begingroup\setlength{\parskip}{\z@}%
202         \@starttoc{\@nameuse{ext@#1}}%
203         \endgroup}}%
204   }{%
205     \RequirePackage{tocbasic}%
206     \PackageInfo{scrhack}{redefining \string\newfloat}%
207     \renewcommand\newfloat[3]{%
208       \ifattoclist{#3}{%
209         \PackageError{scrhack}{extension ‘#3’ already in use}{%
210           Each extension may be used only once.\MessageBreak

```

```

211         You, the class, or another package already uses extension
212         ‘#3’.\MessageBreak
213         \string\newfloat\space command will be ignored!}%
214     }{%
215         \addtotoclist[float]{#3}%
216         \setuptoc{#3}{chapteratlist}%
217         \@namedef{ext@#1}{#3}%
218         \let\float@do=\relax
219         \xdef\@tempa{\noexpand\float@exts{\the\float@exts \float@do{#3}}}%
220         \@tempa
221         \floatplacement{#1}{#2}%
222         \@ifundefined{fname@#1}{\floatname{#1}{#1}}{}%
223         \expandafter\edef\csname ftype@#1\endcsname{\value{float@type}}%
224         \addtocounter{float@type}{\value{float@type}}
225         \restylefloat{#1}%
226         \expandafter\edef\csname fnum@#1\endcsname%
227         {\expandafter\noexpand\csname fname@#1\endcsname{
228             \expandafter\noexpand\csname the#1\endcsname}%
229         \@ifnextchar[%]
230         {\float@newx{#1}}%
231         {\@ifundefined{c@#1}{\newcounter{#1}\@namedef{the#1}{\arabic{#1}}}%
232          {}}}%
233     }%
234     \PackageInfo{scrhack}{redefining \string\listof}%
235     \renewcommand*\listof[2]{%
236         \@ifundefined{ext@#1}{\float@error{#1}}{%
237             \@ifundefined{l@#1}{\expandafter\let\csname l@#1\endcsname\l@figure
238                 \@ifundefined{l@#1}{%
239                     \@namedef{l@#1}{\@dottedtocline{1}{1.5em}{2.3em}}}%
240             }{}%
241             \listoftoc[{#2}]{\csname ext@#1\endcsname}%
242         }%
243     }%
244     \scr@ifexpected{\float@addtolists}{%
245         \long\def\float@addtolists#1{%
246             \def\float@do##1{\addtocontents{##1}{#1} \the\float@exts}%
247         }{%
248             \PackageInfo{scrhack}{undefining \string\float@addtolists}%
249             \let\float@addtolists\relax
250         }{%
251             \PackageWarningNoLine{scrhack}{unkown \string\float@addtolists\space
252                 definition found!\MessageBreak
253                 Maybe you are using a unsupported float version}%
254         }%
255     }{%
256         \PackageWarningNoLine{scrhack}{unknown \string\listof\space
257             definition found!\MessageBreak
258             Maybe you are using a unsupported float version}%
259     }%

```

```

260 }{%
261   \PackageWarningNoLine{scrhack}{unknown \string\newfloat\space
262     definition found!\MessageBreak
263     Maybe you are using a unsupported float version}%
264 }
265 </float & body>

```

## 7.5 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

```

266 <*package & option>
267 \KOMA@ifkey{floatrow}{@scrhack@floatrow}
268 \KOMAExecuteOptions{floatrow=true}
269 </package & option>
270 <*package & body>
271 \AfterPackage*{floatrow}{%
272   \KOMA@key[.scrhack.sty]{floatrow}{%
273     \PackageWarning{scrhack}{option 'floatrow' ignored}%
274     \FamilyKeyStateProcessed
275   }%
276   \if@scrhack@floatrow\scr@hack@load\@pkgextension{floatrow}\fi
277 }
278 </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.

```

279 <*floatrow & body>
280 \scr@ifexpected{\DeclareNewFloatType}{%

```

```

281 \long\def\DeclareNewFloatType#1#2{\def\FB@capttype{#1}%
282 \expandafter\edef\csname ftype@#1\endcsname{\the\c@float@type}%
283 \addtocounter{float@type}{\value{float@type}}%
284 \@namedef{#1name}{#1}\newcounter{#1}%
285 \expandafter\edef\csname fnum@#1\endcsname
286 {\expandafter\noexpand\csname #1name\endcsname\nobreakspace
287 \expandafter\noexpand\csname the#1\endcsname}%
288 \@namedef{the#1}{\arabic{#1}}\flnew@ext{lo#1}\@namedef{fps@#1}{tbp}%
289 \@namedef{l@#1}{\@dottedtocline{1}{1.5em}{2.3em}}%
290 \caption@setkeys[floatrow]{newfloat}{#2}\let\FR@tmp=\relax
291 \xdef\@tempa{\noexpand\flrow@types{\the\flrow@types \FR@tmp{#1}}}%
292 \@tempa}%
293 }{%
294 \scr@ifexpected{\listof}{%
295 \def\listof#1#2{%
296 \ifundefined{ext@#1}{\flrow@error{Unknown float style ‘#1’}}{%
297 \expandafter\providecommand\csname l@#1\endcsname
298 {\@dottedtocline{1}{1.5em}{2.3em}}%
299 \float@listhead{#2}%
300 \begingroup\setlength{\parskip}{\z@}%
301 \starttoc{\@nameuse{ext@#1}}%
302 \endgroup}}%
303 }{%
304 \RequirePackage{tocbasic}%
305 \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!*)

```

306 \renewcommand\DeclareNewFloatType[2]{\def\FB@capttype{#1}%
307 \expandafter\edef\csname ftype@#1\endcsname{\the\c@float@type}%
308 \addtocounter{float@type}{\value{float@type}}%
309 \@namedef{#1name}{#1}\newcounter{#1}%
310 \expandafter\edef\csname fnum@#1\endcsname
311 {\expandafter\noexpand\csname #1name\endcsname\nobreakspace
312 \expandafter\noexpand\csname the#1\endcsname}%
313 \@namedef{the#1}{\arabic{#1}}\flnew@ext{lo#1}\@namedef{fps@#1}{tbp}%
314 \@namedef{l@#1}{\@dottedtocline{1}{1.5em}{2.3em}}%
315 \caption@setkeys[floatrow]{newfloat}{#2}\let\FR@tmp=\relax
316 \xdef\@tempa{\noexpand\flrow@types{\the\flrow@types \FR@tmp{#1}}}%
317 \@tempa
318 \xdef\@tempa{\noexpand\addtotoclist[float]{\@nameuse{ext@FB@capttype}}%
319 \noexpand\setuptoc{\@nameuse{ext@FB@capttype}}{chapteratlist}%
320 }%
321 \@tempa

```

```

322 }%
323 \PackageInfo{scrhack}{redefining \string\listof}%
324 \renewcommand*\listof[2]{%
325   \@ifundefined{ext@#1}{\flow@error{Unknown float style '#1'}}{%
326     \@ifundefined{l@#1}{\expandafter\let\csname l@#1\endcsname\l@figure
327       \@ifundefined{l@#1}{%
328         \@namedef{l@#1}{\@dottedtocline{1}{1.5em}{2.3em}}{}}{%
329       }{}%
330     \listoftoc[{#2}]{\csname ext@#1\endcsname}%
331   }%
332 }%
333 \scr@ifexpected{\float@addtolists}{%
334   \long\def\float@addtolists#1{%
335     \def\float@do##1{\addtocontents{##1}{#1}} \the\float@exts}%
336 }{%
337   \PackageInfo{scrhack}{undefining \string\float@addtolists}%
338   \let\float@addtolists\relax
339 }{%
340   \PackageWarningNoLine{scrhack}{unkown \string\float@addtolists\space
341     definition found!\MessageBreak
342     Maybe you are using a unsupported floatrow version}%
343 }%
344 }{%
345   \PackageWarningNoLine{scrhack}{unknown \string\listof\space
346     definition found!\MessageBreak
347     Maybe you are using a unsupported floatrow version}%
348 }%
349 }{%
350   \PackageWarningNoLine{scrhack}{unknown \string\DeclareNewFloatType\space
351     definition found!\MessageBreak
352     Maybe you are using a unsupported floatrow version}%
353 }
354 </floatrow & body>

```

## 7.6 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`

```
355 <*package & option>
```

```

356 \KOMA@ifkey{listings}{@scrhack@listings}
357 \KOMAExecuteOptions{listings=true}
358 </package & option>
359 <*package & body>
360 \AfterPackage*{listings}{%
361   \KOMA@key[.scrhack.sty]{listings}{%
362     \PackageWarning{scrhack}{option 'listings' ignored}%
363     \FamilyKeyStateProcessed
364   }%
365   \if@scrhack@listings\scr@hack@load\@pkgextension{listings}\fi
366 }
367 </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.

```

368 <*listings & body>
369 \newcommand*{\scr@do@hack@listings}{%
370   \RequirePackage{tocbasic}%
371   \addtotoclist[float]{lol}%
372   \setuptoc{lol}{chapteratlist}%
373   \PackageInfo{scrhack}{redefining \string\lstlistoflistings}%
374   \renewcommand*{\lstlistoflistings}{\listoftoc[{\lstlistlistingname}]{lol}}%
375   \AtBeginDocument{%
376     \scr@ifexpected{\float@addtolists}{%
377       \def\float@addtolists##1{\addtocontents{lol}{##1}}%
378     }{%
379       \PackageInfo{scrhack}{undefining \string\float@addtolists}%
380       \let\float@addtolists\relax
381     }{%
382       \scr@ifexpected{\float@addtolists}{%
383         \def\float@addtolists##1{\addtocontents{lol}{##1}}%
384         \orig@float@addtolists{##1}}%
385       }{%
386         \PackageInfo{scrhack}{setting \string\float@addtolists\MessageBreak
387           to \string\orig@float@addtolists}%
388         \let\float@addtolists\orig@float@addtolists
389       }{%
390         \PackageWarningNoLine{scrhack}{unkown \string\float@addtolists\space
391           definition found!\MessageBreak
392           Maybe you are using a unsupported listings version}%
393       }%
394     }%
395   }%

```



```

396 \let\scr@do@hack@listings\relax
397 }
398 \scr@ifexpected{\lstlistoflistings}{%
399 \def\lstlistoflistings{\bgroup
400 \let\contentsname\lstlistlistingname
401 \let\lst@temp\@starttoc \def\@starttoc##1{\lst@temp{lol}}}%
402 \tableofcontents \egroup}%
403 }{%
404 \scr@do@hack@listings
405 }{%
406 \scr@ifexpected{\lstlistoflistings}{%
407 \def\lstlistoflistings{%
408 \begingroup
409 \ifundefined{@restonecoltrue}{}%
410 \if@twocolumn
411 \@restonecoltrue\onecolumn
412 \else
413 \@restonecolfalse
414 \fi
415 }%
416 \float@listhead{\lstlistlistingname}%
417 \parskip\z@\parindent\z@\parfillskip \z@ \@plus 1fil%
418 \@starttoc{lol}%
419 \ifundefined{@restonecoltrue}{}%
420 \if@restonecol\twocolumn\fi
421 }%
422 \endgroup
423 }%
424 }{%
425 \scr@do@hack@listings
426 }{%
427 \PackageWarningNoLine{scrhack}{unknown \string\lstlistoflistings\space
428 definition found!\MessageBreak
429 Maybe you are using a unsupported listings version}%
430 }%
431 }
432 </listings & body>

```

## 7.7 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 L<sup>A</sup>T<sub>E</sub>X 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`

```
433 <*package & option>
434 \KOMA@ifkey{setspace}{@scrhack@setspace}
435 \KOMAEecuteOptions{setspace=true}
436 </package & option>
437 <*package & body>
438 \AfterPackage*{setspace}{%
439   \KOMA@key[.scrhack.sty]{setspace}{%
440     \PackageWarning{scrhack}{option 'setspace' ignored}%
441     \FamilyKeyStateProcessed
442   }%
443   \if@scrhack@setspace\scr@hack@load\@pkgextension{setspace}\fi
```

```

444 }
445 </package & body>

```

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

```

446 <*setspace & body>
447 \scr@ifexpected{\onehalfspacing}{%
448   \long\def\onehalfspacing{%
449     \setstretch{1.25}% default
450     \ifcase \@ptsize \relax % 10pt
451       \setstretch {1.25}%
452     \or % 11pt
453       \setstretch {1.213}%
454     \or % 12pt
455       \setstretch {1.241}%
456     \fi
457   }%
458 }{%
459   \renewcommand*\onehalfspacing{%
460     \@tempdima=\dimexpr (\f@size pt)*1500/
461                          (\dimexpr \f@baselineskip\relax)*\p@/1000\relax
462     \expandafter\setstretch\expandafter{\strip@pt\@tempdima}%
463   }%
464 }{%
465   \PackageWarning{scrhack}{unknown \string\onehalfspacing\space
466     definition found!\MessageBreak
467     Maybe you are using a unsupported setpace version}%
468 }

```

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

```

469 \scr@ifexpected{\doublespacing}{%
470   \long\def\doublespacing{%
471     \setstretch{1.667}% default
472     \ifcase \@ptsize \relax % 10pt
473       \setstretch {1.667}%
474     \or % 11pt
475       \setstretch {1.618}%
476     \or % 12pt
477       \setstretch {1.655}%
478     \fi
479   }%
480 }{%
481   \renewcommand*\doublespacing{%
482     \@tempdima=\dimexpr (\f@size pt)*2000/
483                      (\dimexpr \f@baselineskip\relax)*\p@/1000\relax

```

```

484     \expandafter\setstretch\expandafter{\strip@pt\@tempdima}%
485   }%
486 }{%
487   \PackageWarning{scrhack}{unknown \string\doublespacing\space
488     definition found!\MessageBreak
489     Maybe you are using a unsupported setpace version}%
490 }
491 </setspace & body>

```

## 7.8 Der lscape-Hack

Das lscape-Paket setzt innerhalb der landscape-Umgebung die Länge `\textheight` auf den Wert von `\textwidth` obwohl es auf der anderen Seite `\textwidth` nicht auf den Wert von `\textheight` setzt. Das ist inkonsequent. Da David Carlisle außerdem angibt, dass `\textwidth` nicht verändert wird, weil das zu Problemen führen konnte, ist es unverständlich, dass `\textheight` verändert wird, obwohl das ebenfalls zu Problemen führen kann, beispielsweise für `showframe` oder `scrpage`. Daher verändere ich die Definition so, dass auch `\textheight` unverändert bleibt. Dabei muss allerdings auch `pdfscape` berücksichtigt werden. Das ist am einfachsten mit `xpatch`.

lscape

```

492 <*package & option>
493 \KOMA@ifkey{lscape}{@scrhack@lscape}%
494 \KOMAExecuteOptions{lscape=true}%
495 </package & option>
496 <*package & body>
497 \AfterPackage*{lscape}{%
498   \if@scrhack@lscape\scr@hack@load\@pkgextension{lscape}\else
499     \KOMA@key[.scrhack.sty]{lscape}{%
500       \PackageWarning{scrhack}{option 'lscape' ignored}%
501       \FamilyKeyStateProcessed
502     }%
503   \fi
504 }
505 </package & body>

```

`\landscape` Über diese Anweisung wird die landscape-Umgebung von `lscape` gestartet. Genau diese muss gepatcht werden. Dafür wird das Paket `xpatch` benötigt. Da der Patch nur geladen wird, wenn die Option dafür gesetzt ist, kann die Option daher nur ein- und ausgeschaltet werden, wenn sie bis zum Laden des Pakets aktiviert wurde.

```

506 <*lscape & body>

```

```

507 \RequirePackage{xpatch}
508 \xpatchcmd{\landscape}{\textheight=\vsize}{%
509   \if@scrhack@lscape

```

Es gibt allerdings in der Tat eine Stelle, an der ein verändertes Wert von `\textheight` benötigt wird. Das ist wenn innerhalb von `\@outputpage` der Wert von `\@colht` reinitialisiert wird. Also wird das entsprechend auch noch hinein gepatcht.

```

510   \scrh@LT@textheight=\vsize
511   \let\scrh@LT@outputpage\@outputpage
512   \def\@outputpage{\scrh@LT@outputpage\global\@colht\scrh@LT@textheight}%
513   \else
514     \textheight=\vsize
515   \fi
516 }{%
517   \PackageInfo{scrhack}{\string\landscape\space patched to make
518     \string\textheight\space change optional}%
519 }{%
520   \PackageWarning{scrhack}{Cannot patch \string\landscape!\MessageBreak
521     Maybe you are using a unsupported lscape version}%
522   \@scrhack@lscapefalse
523 }

```

`\scrh@LT@textheight`

```

524 \newlength{\scrh@LT@textheight}
525 \lscap & body

```

## 7.9 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.

```

526 <*package & option>
527 \KOMAProcessOptions\relax
528 </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.

<b>Symbols</b>	<code>\@spart</code> . . . . . <u>60</u>
<code>\@schapter</code> . . . . .	<u>60</u>

<code>\@ssect</code> .....	60	<b>O</b>	
<b>D</b>		<code>\onehalfspacing</code> .....	446
<code>\DeclareNewFloatType</code> .....	279	Optionen:	
<code>\doublespacing</code> .....	469	<code>floatrow</code> .....	266
<b>F</b>		<code>float</code> .....	164
<code>float</code> (Option) .....	164	<code>hyperref</code> .....	28
<code>\float@addtolists</code> ..	177, 279, 368	<code>listings</code> .....	355
<code>floatrow</code> (Option) .....	266	<code>lscap</code> .....	492
<b>H</b>		<code>setspace</code> .....	433
<code>hyperref</code> (Option) .....	28	<b>S</b>	
<b>L</b>		<code>\scr@do@hack@listings</code> .....	368
<code>\landscape</code> .....	506	<code>\scr@hack@load</code> .....	12
<code>listings</code> (Option) .....	355	<code>\scr@ifexpected</code> .....	1
<code>\listof</code> .....	177, 279	<code>\scrh@LT@textheight</code> (Länge) ..	524
<code>lscap</code> (Option) .....	492	<code>secnumdepth</code> (Zähler) .....	1
<code>\lstlistoflistings</code> .....	368	<code>setspace</code> (Option) .....	433
<b>N</b>		<b>Z</b>	
<code>\newfloat</code> .....	177	Zähler:	
		<code>secnumdepth</code> .....	1

## Change History

v3.03	.....	13	
secnumdepth: erste Version des Pakets .....	5	hyperref: Signalisierung mit \FamilyKeyStateProcessed .....	7
v3.04b			
secnumdepth: Die Reihenfolge von Anweisungen und Optionen grundlegend geändert, um das Paket scrhack unabhängiger von der Reihenfolge beim Laden von Paketen zu machen. ....	5	v3.17	
hyperref: hyperref-Hack wird früher geladen .....	7	\doublespacing: Neu .....	19
v3.12		setspace: Neu .....	18
listings: Signalisierung mit \FamilyKeyStateProcessed .....	15	listings: Defaulteinstellung mit \KOMAEecuteOptions ..	15
float: Signalisierung mit \FamilyKeyStateProcessed .....	10	float: Defaulteinstellung mit \KOMAEecuteOptions ...	10
floatrow: Signalisierung mit \FamilyKeyStateProcessed .....		floatrow: Defaulteinstellung mit \KOMAEecuteOptions ..	13
		\onehalfspacing: Neu .....	19
		hyperref: Defaulteinstellung mit \KOMAEecuteOptions ..	7
		Neuere hyperref-Version deaktiviert ggf. Option hyperref ..	7
		v3.18	
		hyperref: Neuere KOMA-Script-Versionen deaktivieren ggf. Option hyperref .....	7

v3.18a		lscape: Neue Option für neuen	
secnumdepth: Neuer lscape-		Hack .....	20
Hack .....	20		
\landscape: Neu .....	20	\scrh@LT@textheight: neue	
Patchen von \@outputpage .	21	Länge (intern) .....	21