

# The Implementation of the caption kernel\*

Axel Sommerfeldt

`axel.sommerfeldt@f-m.fm`

2011/08/06

## Abstract

The caption kernel consists of two parts – the kernel (`caption3.sty`) and the main package (`caption.sty`).

The kernel provides all the user commands and internal macros which are necessary for typesetting captions and setting parameters regarding these. While the standard  $\text{\LaTeX}$  document classes provide an internal command called `\@makecaption` and no options to control its behavior (except the vertical skips above and below the caption itself), we provide similar commands called `\caption@make` and `\caption@@make`, but with a lot of options which can be selected with `\captionsetup`. Loading the kernel part do not change the output of a  $\text{\LaTeX}$  document – it just provides functionality which can be used by  $\text{\LaTeX} 2_{\epsilon}$  packages which typesets captions, for example the caption and subfig packages.

---

\*This package has version number v1.3b, last revised 2011/08/18.

# Contents

|  |           |
|--|-----------|
| <b>1 Identification</b>  | <b>4</b>  |
| <b>2 Generic helpers</b>   | <b>4</b>  |
| <b>3 Information, Warnings, and Errors</b>                           | <b>8</b>  |
| <b>4 Using the keyval package</b>                                    | <b>8</b>  |
| <b>5 Margin resp. width</b>  | <b>14</b> |
| <b>6 Indentions</b>  | <b>16</b> |
| <b>7 Styles</b>  | <b>17</b> |
| <b>8 Formats</b>   | <b>18</b> |
| <b>9 Label formats</b>   | <b>20</b> |
| <b>10 Label separators</b>   | <b>20</b> |
| <b>11 Text formats</b>   | <b>21</b> |
| <b>12 Fonts</b>  | <b>22</b> |
| <b>13 Justifications</b>   | <b>23</b> |
| 13.1 The ragged2e package . . . . .                                  | 24        |
| <b>14 Vertical spaces before and after captions</b>                  | <b>25</b> |
| <b>15 Positioning</b>  | <b>26</b> |
| <b>16 Hooks</b>  | <b>27</b> |
| <b>17 Lists</b>  | <b>27</b> |
| <b>18 Debug option</b>   | <b>28</b> |
| <b>19 Document classes &amp; Babel support</b>                       | <b>28</b> |
| 19.1 The standard L <sup>A</sup> T <sub>E</sub> X classes . . . . .  | 28        |
| 19.2 The $\mathcal{A}\mathcal{M}\mathcal{S}$ & SMF classes . . . . . | 29        |
| 19.3 The beamer class (Part one) . . . . .                           | 30        |
| 19.4 The KOMA-Script classes . . . . .                               | 32        |
| 19.5 The NTG Dutch classes . . . . .                                 | 32        |
| 19.6 The thesis class . . . . .                                      | 33        |

|           |   |           |
|-----------|---|-----------|
| 19.7      | The frenchb Babel option . . . . .              | 33        |
| 19.8      | The frenchle/pro package . . . . .              | 34        |
| 19.9      | The hungarian and magyar Babel option . . . . . | 35        |
| 19.10     | Unknown document class (or package) . . . . .   | 35        |
| <b>20</b> | <b>Execution of options</b>                     | <b>35</b> |
| <b>21</b> | <b>Making an ‘List of’ entry</b>                | <b>35</b> |
| <b>22</b> | <b>Typesetting the caption</b>                  | <b>36</b> |
| <b>23</b> | <b>Types &amp; sub-types</b>                    | <b>40</b> |
| <b>24</b> | <b>subfig package adaption</b>                  | <b>51</b> |

# 1 Identification

```
1 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
2 \ProvidesPackage{caption3}[2011/08/18 v1.3b caption3 kernel (AR)]
```

# 2 Generic helpers

`\@nameundef` This is the opposite to `\@namedef` which is offered by the  $\text{\LaTeX}$  kernel. We use it to remove the definition of some commands and keyval options after `\begin{document}` (to save  $\text{\TeX}$  memory) and to remove caption options defined with `\captionsetup[⟨type⟩]`.

```
3 \providecommand*\@nameundef[1]{%
4   \expandafter\let\csname #1\endcsname\@undefined}
```

`\l@addto@macro` The  $\text{\LaTeX 2}\epsilon$  kernel offers the internal helper macro `\g@addto@macro` which globally adds tokens to existing macros, like in `\AtBeginDocument`. This is the same but it works local, not global (using `\edef` instead of `\xdef`).

```
5 \providecommand\l@addto@macro[2]{%
6   \begingroup
7     \toks@\expandafter{#1#2}%
8     \edef\@tempa{\endgroup\def\noexpand#1{\the\toks@}}%
9   \@tempa}
```

`\bothIfFirst` `\bothIfFirst` tests if the first argument is not empty, `\bothIfSecond` tests if the second argument is not empty. If yes both arguments get typeset, otherwise none of them.

```
10 \def\bothIfFirst#1#2{%
11   \protected@edef\caption@tempa{#1}%
12   \ifx\caption@tempa\@empty \else
13     #1#2%
14   \fi}
15 \def\bothIfSecond#1#2{%
16   \protected@edef\caption@tempa{#2}%
17   \ifx\caption@tempa\@empty \else
18     #1#2%
19   \fi}
```

`\caption@ifundefined` Similar to `\@ifundefined` offered by the  $\text{\LaTeX}$  kernel, but does not define the undefined macro as `\relax`.

```
20 \newcommand*\caption@ifundefined[1]{%
21   \ifx#1\@undefined
22     \expandafter\@firstoftwo
23   \else\ifx#1\relax
24     \expandafter\expandafter\expandafter\@firstoftwo
25   \else
26     \expandafter\expandafter\expandafter\@secondoftwo
27   \fi\fi}
```

`\caption@ifinlist` This helper macro checks if the first argument is in the comma separated list which is offered as second argument. So for example

```
\caption@ifinlist{frank}{axel, frank, olga, steven}{yes}{no}
```

would expand to yes.

```
28 \newcommand*\caption@ifinlist{%
29   \@expandtwoargs\caption@@ifinlist}
```

```

30 \newcommand*\caption@ifinlist[2]{%
31   \begingroup
32   \def\@tempa##1, #1, ##2\@nil{%
33     \endgroup
34     \ifx\relax##2\relax
35       \expandafter\@secondoftwo
36     \else
37       \expandafter\@firstoftwo
38     \fi}%
39   \@tempa, #2, #1, \@nil}%

\caption@ifin@list \caption@ifin@list{<cmd>}{<list entry>}{<yes>}{<no>}
40 \newcommand*\caption@ifin@list[2]{%
41   \caption@ifempty@list#1%
42   {\@secondoftwo}%
43   {\@expandtwoargs\caption@ifin@list{#2}{#1}}

\caption@g@addto@list \caption@g@addto@list{<cmd>}{<list entry>}
44 \newcommand*\caption@g@addto@list[2]{%
45   \caption@ifempty@list#1{\gdef#1{#2}}{\g@addto@macro#1{, #2}}

\caption@l@addto@list \caption@l@addto@list{<cmd>}{<list entry>}
46 \newcommand*\caption@l@addto@list[2]{%
47   \caption@ifempty@list#1{\def#1{#2}}{\l@addto@macro#1{, #2}}

\caption@g@removefrom@list \caption@g@removefrom@list{<cmd>}{<list entry>}
48 \newcommand*\caption@g@removefrom@list[2]{%
49   \caption@l@removefrom@list#1{#2}%
50   \global\let#1#1}

\caption@l@removefrom@list \caption@l@removefrom@list{<cmd>}{<list entry>}
Caveat: <cmd> will be expanded during this process since \@removeelement is using \edef
to build the new list!
51 \newcommand*\caption@l@removefrom@list[2]{%
52   \caption@ifempty@list#1{\@expandtwoargs\@removeelement{#2}#1#1}}

\caption@for@list \caption@for@list{<cmd>}{<code with #1>}
53 \newcommand*\caption@for@list[2]{%
54   \caption@ifempty@list#1{}%
55   \def\caption@tempb##1{#2}%
56   \@for\caption@tempa:=#1\do{%
57     \expandafter\caption@tempb\expandafter{\caption@tempa}}}}

\caption@ifempty@list \caption@ifempty@list{<cmd>}{<true>}{<false>}
58 \newcommand*\caption@ifempty@list[1]{%
59   \ifx#1\@undefined
60     \expandafter\@firstoftwo
61   \else\ifx#1\relax
62     \expandafter\expandafter\expandafter\@firstoftwo
63   \else\ifx#1\@empty
64     \expandafter\expandafter\expandafter\expandafter
65     \expandafter\expandafter\expandafter\@firstoftwo
66   \else

```

```

67 \expandafter\expandafter\expandafter\expandafter
68 \expandafter\expandafter\expandafter\@secondoftwo
69 \fi\fi\fi}

```

For setting and testing boolean options we offer these three helper macros:

```

\caption@setbool
\caption@set@bool
\caption@ifbool
\caption@undefbool

\caption@setbool{<name>}{<value>}
      (with value = false/true/no/yes/off/on/0/1)
\caption@ifbool{<name>}{<if-clause>}{<else-clause>}
\caption@undefbool{<name>}

70 \newcommand*\caption@setbool[1]{%
71 \expandafter\caption@set@bool\csname caption@if#1\endcsname}

72 \newcommand*\caption@set@bool[2]{%
73 \caption@ifinlist{#2}{1,true,yes,on}{%
74 \let#1\@firstoftwo
75 }{\caption@ifinlist{#2}{0,false,no,off}{%
76 \let#1\@secondoftwo
77 }{%
78 \caption@Error{Undefined boolean value `#2'}%
79 }}}

80 \newcommand*\caption@ifbool[1]{\@nameuse{caption@if#1}}
81 \newcommand*\caption@undefbool[1]{\@nameundef{caption@if#1}}

\caption@teststar
\caption@teststar{<cmd>}{<star arg>}{<non-star arg>}
\caption@teststar@{<cmd>}{<star arg>}{<non-star arg>}

82 \newcommand*\caption@teststar[3]{\@ifstar{#1{#2}}{#1{#3}}}

83 \newcommand*\caption@teststar@[3]{%
84 \@ifstar{#1{#2}}{\caption@ifatletter{#1{#2}}{#1{#3}}}}
85 \AtBeginDocument{\let\caption@teststar@\caption@teststar}

86 \newcommand*\caption@ifatletter{%
87 \ifnum\the\catcode`\@=11
88 \expandafter\@firstoftwo
89 \else
90 \expandafter\@secondoftwo
91 \fi}
92 \AtBeginDocument{\let\caption@ifatletter\@secondoftwo}

\caption@withoptargs
\caption@withoptargs{<cmd>}

93 \newcommand*\caption@withoptargs[1]{%
94 \@ifstar
95 {\def\caption@tempa{*}\caption@@withoptargs#1}%
96 {\def\caption@tempa{}\caption@@withoptargs#1}}

97 \def\caption@@withoptargs#1{%
98 \@ifnextchar[%]
99 {\caption@@@withoptargs#1}%
100 {\caption@@@withoptargs#1}}

101 \def\caption@@@withoptargs#1[#2]{%
102 \l@addto@macro\caption@tempa{[#{#2}]}%
103 \caption@@withoptargs#1}

```

```

104 \def\caption@@@withoptargs#1{%
105   \expandafter#1\expandafter{\caption@tempa}}

```

`\caption@CheckCommand` `\caption@CheckCommand{<macro>}{<definition of macro>}`  
`\caption@IfCheckCommand` checks if a command already exists, with the same definition. It can be used more-than-once to check if one of multiple definitions will finally match. (It redefines itself later on to `\@gobbletwo` if the two commands match fine, making further checks harmless.)  
`\caption@IfCheckCommand{<true>}{<false>}`  
 will execute the *<true>* code if one match was finally given, the *<false>* code otherwise. (It simply checks if `\caption@CheckCommand` is `\@gobbletwo` and restores the starting definition of `\caption@CheckCommand`.)

```

106 \newcommand\caption@DoCheckCommand[2]{%
107   \begingroup
108     \let\@tempa#1%
109     #2%
110     \ifx\@tempa#1%
111       \endgroup
112       \let\caption@CheckCommand\@gobbletwo
113     \else
114       \endgroup
115     \fi}
116 \@onlypreamble\caption@DoCheckCommand

117 \let\caption@CheckCommand\caption@DoCheckCommand
118 \@onlypreamble\caption@CheckCommand

119 \newcommand*\caption@IfCheckCommand{%
120   \ifx\caption@CheckCommand\@gobbletwo
121     \let\caption@CheckCommand\caption@DoCheckCommand
122     \expandafter\@firstoftwo
123   \else
124     \expandafter\@secondoftwo
125   \fi}
126 \@onlypreamble\caption@IfCheckCommand

```

`\caption@AtBeginDocument` `\caption@AtBeginDocument*{<code>}`  
 Same as `\AtBeginDocument` but the execution of code will be surrounded by two `\PackageInfos`. The starred variant causes the code to be executed after all code specified using the non-starred variant.

```

127 \let\caption@begindocumenthook\@empty
128 \let\caption@@begindocumenthook\@empty

129 \def\caption@AtBeginDocument{%
130   \caption@teststar@g@addto@macro
131   \caption@@begindocumenthook\caption@begindocumenthook}
132 %\@onlypreamble\caption@AtBeginDocument

133 \AtBeginDocument{%
134   \caption@InfoNoLine{Begin \noexpand\AtBeginDocument code}%

135   \def\caption@AtBeginDocument{%
136     \@ifstar{\g@addto@macro\caption@@begindocumenthook}\@firstofone}%
137   \caption@begindocumenthook
138   \let\caption@begindocumenthook\relax

```

```

139 \def\caption@AtBeginDocument{%
140 \ifstar\@firstofone\@firstofone}%
141 \caption@@begindocumenthook
142 \let\caption@@begindocumenthook\relax
143 \caption@InfoNoLine{End \noexpand\AtBeginDocument code}}

```

### 3 Information, Warnings, and Errors

```

\caption@Info \caption@Info{<message>}
144 \newcommand*\caption@Info[1]{%
145 \PackageInfo{caption}{#1}}

```

```

\caption@InfoNoLine \caption@InfoNoLine{<message>}

```

*Note:* The \@gobble at the end of the 2nd argument of \PackageInfo suppresses the line number info. See TLC2[?], A.4.7, p885 for details.

```

146 \newcommand*\caption@InfoNoLine[1]{%
147 \PackageInfo{caption}{#1\@gobble}}

```

```

\caption@Warning \caption@Warning{<message>}
148 \newcommand*\caption@Warning[1]{%
149 \caption@WarningNoLine{#1\on@line}}

```

```

\caption@WarningNoLine \caption@WarningNoLine{<message>}
150 \newcommand*\caption@WarningNoLine[1]{%
151 \PackageWarning{caption}{#1.^~J\caption@wh\@gobbletwo}}
152 \newcommand*\caption@wh{%
153 See the caption package documentation for explanation.}

```

```

\caption@Error \caption@Error{<message>}
154 \newcommand*\caption@Error[1]{%
155 \PackageError{caption}{#1}\caption@eh}
156 \newcommand*\caption@eh{%
157 If you do not understand this error, please take a closer look\MessageBreak
158 at the documentation of the 'caption' package, especially the\MessageBreak
159 section about errors.\MessageBreak\@ehc}

```

```

\caption@KV@err
160 \let\caption@KV@err\caption@Error

```

### 4 Using the keyval package

We need the keyval package for option handling, so we load it here.

```

161 \RequirePackage{keyval}[1997/11/10]

```

```

\undefine@key \undefine@key{<family>}{<key>}

```

This helper macro is the opposite of \define@key, it removes a keyval definition.

```

162 \providecommand*\undefine@key[2]{%
163 \@nameundef{KV@#1@#2}\@nameundef{KV@#1@#2@default}}

```



```

\@onlypreamble@key \onlypreamble@key{<family>}{<key>}
Analogous to \@onlypreamble from LATEX 2ε.
164 \providecommand*\@preamble@keys{}
165 \providecommand*\@onlypreamble@key[2]{\@cons\@preamble@keys{{#1}{#2}}}
166 \@onlypreamble\@onlypreamble@key
167 \@onlypreamble\@preamble@keys

168 \providecommand*\@notprerr@key[1]{\KV@err{Can be used only in preamble}}

169 \caption@AtBeginDocument*{%
170   \def\@elt#1#2{\expandafter\let\csname KV@#1@#2\endcsname\@notprerr@key}%
171   \@preamble@keys
172   \let\@elt\relax}

\DeclareCaptionOption \DeclareCaptionOption{<option>}[<default value>]{<code>}
\DeclareCaptionOption*{<option>}[<default value>]{<code>}
We declare our options using these commands (instead of using \DeclareOption
offered by LATEX 2ε), so the keyval package is used. The starred form makes the op-
tion available during the lifetime of the current package only, so they can be used with
\usepackage, but not with \captionsetup later on.
173 \newcommand*\DeclareCaptionOption{%
174   \caption@teststar\caption@declareoption\AtEndOfPackage\@gobble}
175 \@onlypreamble\DeclareCaptionOption

176 \newcommand*\caption@declareoption[2]{%
177   #1{\undefine@key{caption}{#2}}\define@key{caption}{#2}}
178 \@onlypreamble\caption@declareoption

\DeclareCaptionOptionNoValue \DeclareCaptionOptionNoValue{<option>}{<code>}
\DeclareCaptionOptionNoValue*{<option>}{<code>}
Same as \DeclareCaptionOption but issues an error if a value is given.
179 \newcommand*\DeclareCaptionOptionNoValue{%
180   \caption@teststar\caption@declareoption@novalue\AtEndOfPackage\@gobble}
181 \@onlypreamble\DeclareCaptionOptionNoValue

182 \newcommand\caption@declareoption@novalue[3]{%
183   \caption@declareoption{#1}{#2}[\KV@err]{%
184     \caption@option@novalue{#2}{{#1}{#3}}}
185 \@onlypreamble\caption@declareoption@novalue

186 \newcommand*\caption@option@novalue[2]{%
187   \ifx\KV@err#2%
188     \expandafter\@firstofone
189   \else
190     \KV@err{No value allowed for #1}%
191     \expandafter\@gobble
192   \fi}

\ifcaptionsetup@star If the starred form of \captionsetup is used, this will be set to true. (It will be reset
to false at the end of \caption@setkeys.)
193 \newif\ifcaptionsetup@star

\captionsetup \captionsetup[<type>]{<keyval-list of options>}
\captionsetup*{<type>}{<keyval-list of options>}

```

If the optional argument ‘type’ is specified, we simply save or append the option list, otherwise we ‘execute’ it with `\setkeys`. (The non-starred variant issues a warning if *⟨keyval-list of options⟩* is not used later on.)

*Note:* The starred variant will be used inside packages automatically.

```

194 \newcommand*\captionsetup{%
195   \caption@teststar@\@captionsetup\@gobble\@firstofone}

196 \newcommand*\@captionsetup[1]{%
197   \captionsetup@startrue#1\captionsetup@starfalse
198   \ifnextchar[\caption@setup@options\caption@setup}

199 \newcommand*\caption@setup{\caption@setkeys{caption}}

200 \def\caption@setup@options[#1]#2{%
201   \@bsphack
202   \ifcaptionsetup@star\captionsetup@starfalse\else\caption@addtooptlist{#1}\fi
203   \expandafter\caption@l@addto@list\csname caption@opt@#1\endcsname{#2}%
204   \@esphack}

```

```

\clearcaptionsetup \clearcaptionsetup[⟨option⟩]{⟨type⟩}
\clearcaptionsetup* [⟨option⟩]{⟨type⟩}

```

This removes the saved option list associated with *⟨type⟩*. If *⟨option⟩* is given, only this option will be removed from the list. (The starred variant does not issue warnings.)

*Note:* The starred variant will be used inside packages automatically.

```

205 \newcommand*\clearcaptionsetup{%
206   \caption@teststar@\@clearcaptionsetup\@gobble\@firstofone}

207 \newcommand*\@clearcaptionsetup[1]{%
208   \let\caption@tempa#1%
209   \@testopt\@clearcaptionsetup{}}

210 \def\@clearcaptionsetup[#1]#2{%
211   \@bsphack
212   \expandafter\caption@ifempty@list\csname caption@opt@#2\endcsname
213   {\caption@tempa{\caption@Warning{Option list ‘#2’ undefined}}}%
214   {\ifx,#1,%
215     \caption@clearsetup{#2}%
216   \else
217     \caption@@removefromsetup{#1}{#2}%
218   \fi}%
219   \@esphack}

220 \newcommand*\caption@clearsetup[1]{%
221   \caption@removefromoptlist{#1}%
222   \@nameundef{caption@opt@#1}}

223 \newcommand*\caption@removefromsetup{%
224   \let\caption@tempa\@gobble
225   \caption@@removefromsetup}

226 \newcommand*\caption@@removefromsetup[2]{%
227   \expandafter\let\expandafter\@tempa\csname caption@opt@#2\endcsname
228   \expandafter\let\csname caption@opt@#2\endcsname\@undefined
229   \def\@tempb##1=##2\@nil{##1}%
230   \edef\@tempc{#1}%
231   \@for\@tempa:=\@tempa\do{%
232     \edef\@tempd{\expandafter\@tempb\@tempa=\@nil}%

```

```

233 \ifx\@tempd\@tempc
234 \let\caption@tempa\@gobble
235 \else
236 \expandafter\expandafter\expandafter\caption@l@addto@list
237 \expandafter\csname caption@opt@#2\expandafter\endcsname
238 \expandafter{\@tempa}%
239 \fi}%
240 \expandafter\caption@ifempty@list\csname caption@opt@#2\endcsname
241 {\caption@removefromoptlist{#2}}}%
242 \caption@tempa{\caption@Warning{%
243 Option `#1' was not in list `#2'\MessageBreak}}}
```

`\showcaptionsetup` `\showcaptionsetup[<package>]{<type>}`

This comes for debugging issues: It shows the saved option list which is associated with *<type>*.

```

244 \newcommand*\showcaptionsetup[2][\@firstofone]{%
245 \@bsphack
246 \GenericWarning{ }{%
247 #1 Caption Info: Option list on `#2'\MessageBreak
248 #1 Caption Data: \@ifundefined{caption@opt@#2}{%
249 -none-%
250 }{%
251 {\expandafter\expandafter\expandafter\strip@prefix
252 \expandafter\meaning\csname caption@opt@#2\endcsname}%
253 }%
254 \@esphack}

255 \DeclareCaptionOption{options}{\caption@setoptions{#1}}
256 \DeclareCaptionOption{options*}{\caption@setoptions*{#1}}
```

`\caption@setoptions` `\caption@setoptions*{<type or environment or...>}`

Caption options which have been saved with `\captionsetup[<type>]` can be executed by using this command. It simply executes the saved option list (and clears it afterwards), if there is any. (The starred variant do not clear the option list.)

```

257 \newcommand*\caption@setoptions{%
258 \caption@teststar\caption@setoptions\@gobble\@firstofone}

259 \newcommand*\caption@setoptions[2]{%
260 \caption@Debug{options=#2}%
261 \expandafter\let\expandafter\caption@opt\csname caption@opt@#2\endcsname
262 \ifx\caption@opt\relax \else
263 \caption@xsetup\caption@opt
264 #1{\caption@clearsetup{#2}}% #1 = \@firstofone -or- \@gobble
265 \fi}

266 \newcommand*\caption@xsetup[1]{\expandafter\caption@setup\expandafter{#1}}
```

`\caption@addtooptlist` `\caption@addtooptlist{<type>}`

`\caption@removefromoptlist` `\caption@removefromoptlist{<type>}`

Adds or removes an *<type>* to the list of unused caption options. Note that the catcodes of *<type>* are sanitized here so removing *<type>* from the list do not fail when the float package is used (since `\float@getstyle` gives a result which tokens have catcode 12 = “other”).

```

267 \newcommand*\caption@addtooptlist[1]{%
```

```

268 \ifundefined{caption@opt@#1@lineno}{%
269   \caption@dooptlist\caption@g@addto@list{#1}%
270   \expandafter\xdef\csname caption@opt@#1@lineno\endcsname{\the\inputlineno}%
271 }{}}

272 \newcommand*\caption@removefromoptlist[1]{%
273   \caption@dooptlist\caption@g@removefrom@list{#1}%
274   \global\expandafter\let\csname caption@opt@#1@lineno\endcsname\@undefined}

275 \newcommand*\caption@dooptlist[2]{%
276   \begingroup
277   \edef\@tempa{#2}\@onelevel@sanitize\@tempa
278   \expandafter#1\expandafter\caption@optlist\expandafter{\@tempa}%
279   \endgroup}

280 \AtEndDocument{%
281   \caption@for@list\caption@optlist{%
282     \caption@WarningNoLine{%
283       Unused \string\captionsetup[#1]
284       on input line \csname caption@opt@#1@lineno\endcsname}}}%

```

`\caption@setkeys` `\caption@setkeys[<package>]{<family>}{<key-values>}`

This one simply calls `\setkeys{<family>}{<key-values>}` but lets the error messages not refer to the keyval package, but to the *<package>* package instead.

```

285 \newcommand*\caption@setkeys{\@dblarg\caption@@setkeys}

286 \long\def\caption@@setkeys[#1]#2#3{%
287   \@bsphack

288   \expandafter\let\csname ORI@KV@err\caption@keydepth\endcsname\KV@err
289   \expandafter\let\csname ORI@KV@errx\caption@keydepth\endcsname\KV@errx
290   \expandafter\let\expandafter\KV@err\csname #1@KV@err\endcsname
291   \ifx\KV@err\relax
292     \def\KV@err##1{\PackageError{#1}{##1}{%
293       See the #1 package documentation for explanation.}}%
294   \fi
295   \let\KV@errx\KV@err
296   \edef\caption@keydepth{\caption@keydepth i}%

297   \caption@Debug{\protect\setkeys{#2}{#3}}%
298   \setkeys{#2}{#3}%

299   \edef\caption@keydepth{\expandafter\@gobble\caption@keydepth}%
300   \expandafter\let\expandafter\KV@err\csname ORI@KV@err\caption@keydepth\endcsname
301   \expandafter\let\expandafter\KV@errx\csname ORI@KV@errx\caption@keydepth\endcsname

302   \ifx\caption@keydepth\@empty \captionsetup@starfalse \fi

303   \@esphack}

304 \let\caption@keydepth\@empty

```

`\caption@ExecuteOptions` `\caption@ExecuteOptions{<package>}{<key-values>}`

We execute our options using the keyval interface, so we use this one instead of `\ExecuteOptions` offered by L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>.

```

305 \newcommand*\caption@ExecuteOptions[2]{%
306   \expandafter\@expandtwoargs\csname caption@setkeys@#1\endcsname{#1}{#2}}%
307 \@onlypreamble\caption@ExecuteOptions

```

`\caption@ProcessOptions` `\caption@ProcessOptions*{<package>}`

We process our options using the keyval package, so we use this one instead of `\ProcessOptions` offered by L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>. The starred variant do not process the global options. (This code was taken from the hyperref package[2] v6.74 and improved.)

```

308 \newcommand*\caption@ProcessOptions{%
309   \caption@teststar\caption@@ProcessOptions\@gobble\@firstofone}
310 \@onlypreamble\caption@ProcessOptions

311 \newcommand\caption@@ProcessOptions[2]{%
312   \let\@tempc\relax
313   \let\caption@tempa\@empty
314   #1{% \@firstofone -or- \@gobble
315     \@for\CurrentOption:=\@classoptionslist\do{%
316       \ifundefined{KV@#2@\CurrentOption}{}{%
317         \ifundefined{KV@#2@\CurrentOption @default}{%
318           \PackageInfo{#2}{Global option '\CurrentOption' ignored}%
319         }{%
320           \PackageInfo{#2}{Global option '\CurrentOption' processed}%
321           \edef\caption@tempa{\caption@tempa\CurrentOption,%
322             \@expandtwoargs\@removeelement\CurrentOption
323             \@unusedoptionlist\@unusedoptionlist
324           }%
325         }%
326       }%
327       \let\CurrentOption\@empty
328     }%
329   \caption@ExecuteOptions{#2}{\caption@tempa\@optionlist{\@currname.\@current}}%
330   \AtEndOfPackage{\let\@unprocessedoptions\relax}}
331 \@onlypreamble\caption@@ProcessOptions

```

`\caption@SetupOptions` `\caption@SetupOptions{<package>}{<code>}`

After calling this macro `\caption@ExecuteOptions` and `\usepackage[<options>]{<package>}` will both be mapped to `<code>` with `<package>` and `<options>` as arguments #1 and #2. (This helps avoiding “Option clash” errors.)

```

332 \newcommand*\caption@packagelist{}
333 \@onlypreamble\caption@packagelist

334 \newcommand\caption@SetupOptions[2]{%
335   \@namedef{caption@setkeys@#1}##1##2{#2}%
336   \expandafter\@onlypreamble\csname caption@setkeys@#1\endcsname
337   \@cons\caption@packagelist{{#1}}
338 \@onlypreamble\caption@SetupOptions

339 \let\caption@onefilewithoptions\@onefilewithoptions
340 \def\@onefilewithoptions#1[#2]{%
341   \begingroup
342   \def\@tempa{%
343     \endgroup
344     \caption@onefilewithoptions{#1}{{#2}}}%
345   \def\@tempb{#1}%
346   \def\@elt##1{%
347     \def\@tempc{##1}%
348     \ifx\@tempb\@tempc
349       \def\@tempa{%
350         \endgroup

```

```

351         \caption@ExecuteOptions{#1}{#2}%
352         \caption@onefilewithoptions{#1}[]}%
353     \fi}
354 \caption@packagelist
355 \@tempa}
356 \@onlypreamble\caption@onefilewithoptions

```

## 5 Margin resp. width

`\captionmargin` `\captionwidth` contain the extra margin resp. the total width used for captions. Please never set these values in a direct way, they are just accessible in user documents to provide compatibility to *vI.x*.

Note that we can only set one value at a time, ‘margin’ *or* ‘width’. If `\captionwidth` is not zero we will take this value afterwards, otherwise `\captionmargin` and `\captionmargin@`.

```

357 \newdimen\captionmargin
358 \newdimen\captionmargin@
359 \newdimen\captionwidth

360 \DeclareCaptionOption{margin}{\setcaptionmargin{#1}}
361 \DeclareCaptionOption{margin*}{\setcaptionmargin*{#1}}
362 \DeclareCaptionOption{width}{\setcaptionwidth{#1}}
363 \DeclareCaptionOption{width*}{\setcaptionwidth*{#1}}

364 \DeclareCaptionOption{calcmargin}{\caption@setcalcmargin{#1}}
365 \DeclareCaptionOption{calcmargin*}{\caption@setcalcmargin*{#1}}
366 \DeclareCaptionOption{calcwidth}{\caption@setcalcwidth{#1}}
367 \DeclareCaptionOption{calcwidth*}{\caption@setcalcwidth*{#1}}

368 \DeclareCaptionOption{twoside}[1]{\caption@set@bool\caption@iftwoside{#1}}
369 \DeclareCaptionOptionNoValue{oneside}{\caption@set@bool\caption@iftwoside0}

370 \DeclareCaptionOption{minmargin}{\caption@setoptcmd\caption@minmargin{#1}}
371 \DeclareCaptionOption{maxmargin}{\caption@setoptcmd\caption@maxmargin{#1}}

```

`\setcaptionmargin` `\setcaptionmargin{<amount>}`  
`\setcaptionmargin*{<amount>}`

Please never use this in user documents, it’s just there to provide compatibility to the `caption2` package.

```

372 \newcommand*\setcaptionmargin{%
373     \caption@resetcalcmargin
374     \caption@setmargin}

375 \newcommand*\caption@setmargin{%
376     \caption@teststar\caption@@setmargin\@gobble\@firstofone}

377 \newcommand*\caption@@setmargin[2]{%
378     #1{\captionwidth\z@}%
379     \caption@@@setmargin#2,#2,\@nil}

380 \def\caption@@@setmargin#1,#2,#3\@nil{%
381     \setlength\captionmargin@{#2}%
382     \setlength\captionmargin{#1}%
383     \addtolength\captionmargin@{-\captionmargin}}

```

`\setcaptionwidth` `\setcaptionwidth{<amount>}`  
`\setcaptionwidth*{<amount>}`  
Please never use this in user documents, it's just there to provide compatibility to the `caption2` package.

```
384 \newcommand*\setcaptionwidth{%
385   \caption@resetcalcmargin
386   \caption@setwidth}

387 \newcommand*\caption@setwidth{%
388   \caption@teststar\caption@@setwidth\@gobble\@firstofone}

389 \newcommand*\caption@@setwidth[2]{%
390   #1{\captionmargin\z@\captionmargin@\z@}%
391   \setlength\captionwidth{#2}}%
```

`\caption@resetcalcmargin`

```
392 \newcommand*\caption@resetcalcmargin{%
393   \let\caption@calcmargin@hook\@empty}
```

`\caption@setcalcmargin`

```
394 \newcommand*\caption@setcalcmargin{%
395   \caption@teststar{\caption@@setcalcmargin\caption@setmargin}%
396   \@secondoftwo\@firstoftwo}

397 \newcommand*\caption@@setcalcmargin[3]{%
398   #2{\caption@resetcalcmargin
399     \l@addto@macro\caption@calcmargin@hook{#1{#3}}}%
400     {\l@addto@macro\caption@calcmargin@hook{#1*{#3}}}}
```

`\caption@setcalwidth`

```
401 \newcommand*\caption@setcalwidth{%
402   \caption@teststar{\caption@@setcalcmargin\caption@setwidth}%
403   \@secondoftwo\@firstoftwo}
```

`\caption@counter` This counter numbers the captions. At the moment it will be used inside `\caption@ifoddpage` only.

```
404 \newcommand*\caption@thecounter{0}

405 \newcommand*\caption@stepcounter{%
406   \@tempcnta\caption@thecounter
407   \advance\@tempcnta\@ne
408   \xdef\caption@thecounter{\the\@tempcnta}}
```

`\caption@newlabel` This command is a modified version of `\newlabel` from L<sup>A</sup>T<sub>E</sub>X2<sub>ε</sub>. It will be written to the `.aux` file to pass label information from one run to another. (We use it inside `\caption@ifoddpage` and `\caption@ragged`.)

```
409 \newcommand*\caption@newlabel{\@newl@bel\caption@r}
```

`\caption@thepage` This command is a modified version of `\thepage` from L<sup>A</sup>T<sub>E</sub>X2<sub>ε</sub>. It will be used inside `\caption@ifoddpage` only.

```
410 \newcommand*\caption@thepage{\the\c@page}
```

`\caption@label` This command is a modified version of `\label` from L<sup>A</sup>T<sub>E</sub>X2e. It will be used inside `\caption@ifoddpage` and `\FP@helpNote`.

```

411 \newcommand*\caption@label[1]{%
412   \caption@@label
413   \protected@write\@auxout{\let\caption@thepage\relax}%
414     {\string\caption@newlabel{#1}{\caption@thepage}}}
415 \newcommand*\caption@@label{%
416   \global\let\caption@@label\relax
417   \protected@write\@auxout{}%
418     {\string\providecommand*\string\caption@newlabel[2]{}}}
```

`\caption@pageref` This command is a modified version of `\pageref` from L<sup>A</sup>T<sub>E</sub>X2e. It will be used inside `\caption@ifoddpage` and `\FP@helpNote`.

```

419 \newcommand*\caption@pageref[1]{%
420   \expandafter\ifx\csname caption@r@#1\endcsname\relax
421     \G@refundefinedtrue % => 'There are undefined references.'
422     \@latex@warning{Reference '#1' on page \thepage \space undefined}%
423   \else
424     \expandafter\let\expandafter\caption@thepage\csname caption@r@#1\endcsname
425   \fi}
```

`\caption@ifoddpage` At the moment this macro uses an own label...ref mechanism, but an alternative implementation method would be using the `refcount` package[3] and `\ifodd\getpagerefnumber{...}`.  
*Note:* This macro re-defines itself so the `.aux` file will only be used once per group.

```

426 \newcommand*\caption@ifoddpage{%
427   \caption@iftwoside{%
428     \caption@label\caption@thecounter
429     \caption@pageref\caption@thecounter
430     \ifodd\caption@thepage
431       \let\caption@ifoddpage\@firstoftwo
432     \else
433       \let\caption@ifoddpage\@secondoftwo
434     \fi
435   }{\let\caption@ifoddpage\@firstoftwo}%
436   \caption@ifoddpage}
```

`\caption@setoptcmd` `\caption@setoptcmd{<cmd>}{<off-or-value>}`

```

437 \newcommand*\caption@setoptcmd[2]{%
438   \caption@ifinlist{#2}{0,false,no,off}{\let#1\@undefined}{\def#1{#2}}}
```

## 6 Indentions

`\caption@indent` These are the indentions we support.

```

\caption@parindent 439 \newdimen\caption@indent
\caption@hangindent 440 \newdimen\caption@parindent
441 \newdimen\caption@hangindent

442 \DeclareCaptionOption{indent}[\leftmargini]{% obsolete!
443   \setlength\caption@indent{#1}}
444 \DeclareCaptionOption{indentation}[\leftmargini]{%
445   \setlength\caption@indent{#1}}
```



```

446 \DeclareCaptionOption{parindent}{%
447     \setlength\caption@parindent{#1}}
448 \DeclareCaptionOption{hangindent}{%
449     \setlength\caption@hangindent{#1}}
450 \DeclareCaptionOption{parskip}{%
451     \l@addto@macro\caption@@par{\setlength\parskip{#1}}}

```

There is an option clash between the KOMA-Script document classes and the caption kernel, both define the options `parindent` and `parskip` but with different meaning. Furthermore the ones defined by the caption kernel take a value as parameter but the KOMA-Script ones do not. So we need special versions of the options `parindent` and `parskip` here which determine if a value is given (and therefore should be treated as our option) or not (and therefore should be ignored by us).<sup>1</sup>

```

452 \providecommand*\caption@ifkomaclass{%
453     \caption@ifundefined\scr@caption\@gobble\@firstofone}
454 \@onlypreamble\caption@ifkomaclass
455 \caption@ifkomaclass{%
456     \let\caption@KV@parindent\KV@caption@parindent
457     \DeclareCaptionOption{parindent}[]{%
458         \ifx,#1,%
459             \caption@Debug{Option 'parindent' ignored}%
460         \else
461             \caption@KV@parindent{#1}%
462         \fi}%
463     \let\caption@KV@parskip\KV@caption@parskip
464     \DeclareCaptionOption{parskip}[]{%
465         \ifx,#1,%
466             \caption@Debug{Option 'parskip' ignored}%
467         \else
468             \caption@KV@parskip{#1}%
469         \fi}%
470 }

```

## 7 Styles

```

\DeclareCaptionStyle \DeclareCaptionStyle{<name>}[<single-line-list-of-KV>]{<list-of-KV>}
471 \newcommand*\DeclareCaptionStyle[1]{%
472     \@testopt{\caption@declarestyle{#1}}{}}
473 \@onlypreamble\DeclareCaptionStyle
474 \def\caption@declarestyle#1[#2]#3{%
475     \global\@namedef{caption@sls@#1}{#2}%
476     \global\@namedef{caption@sty@#1}{#3}}
477 \@onlypreamble\caption@declarestyle
478 \DeclareCaptionOption{style}{\caption@setstyle{#1}}
479 \DeclareCaptionOption{style*}{\caption@setstyle*{#1}}
480 \DeclareCaptionOption{singlelinecheck}[1]{\caption@set@bool\caption@ifslc{#1}}
481 \DeclareCaptionOption{slc}[1]{\KV@caption@singlelinecheck{#1}}

```

---

<sup>1</sup>This problem was completely solved due a change of `\caption@ProcessOptions` in the caption kernel *v1.0h*, but we still need this workaround since these options would otherwise still collide with the current version 1.3 of the subfig package (Sigh!)

```
\caption@setstyle \caption@setstyle{<name>}
\caption@setstyle*{<name>}
```

Selecting a caption style means saving the additional *<single-line-list-of-KV>* (this will be done by `\caption@sls`), resetting the caption options to the base ones (this will be done using `\caption@resetstyle`) and executing the *<list-of-KV>* options (this will be done using `\caption@setup`).

The starred version will give no error message if the given style is not defined.

```
482 \newcommand*\caption@setstyle{%
483   \caption@teststar\caption@@setstyle\@gobble\@firstofone}

484 \newcommand*\caption@@setstyle[2]{%
485   \ifundefined{caption@sty@#2}%
486     {#1{\caption@Error{Undefined style `#2'}}}%
487     {\expandafter\let\expandafter\caption@sty\csname caption@sty@#2\endcsname
488     \ifx\caption@setstyle@flag\@undefined
489       \let\caption@setstyle@flag\relax
490       \caption@resetstyle
491       \caption@xsetup\caption@sty
492       \let\caption@setstyle@flag\@undefined
493     }else
494       \caption@xsetup\caption@sty
495     \fi
496     \expandafter\let\expandafter\caption@sls\csname caption@sls@#2\endcsname
497     \expandafter\caption@l@addto@list\expandafter\caption@opt@singleline
498     \expandafter{\caption@sls}}}
```

`\caption@resetstyle` This resets (nearly) all caption options to the base ones. *Note that this does not touch the skips and the positioning!*

```
499 \newcommand*\caption@resetstyle{%
500   \caption@setup{%
501     format=plain,labelformat=default,labelsep=colon,textformat=simple,%
502     justification=justified,font=,size=,labelfont=,textfont=,%
503     margin=0pt,minmargin=0,maxmargin=0,%
504     indent=0pt,parindent=0pt,hangindent=0pt,%
505     slc,rule,strut}%
506   \caption@clearsetup{singleline}}}
```

Currently there are two pre-defined styles, called ‘base’ & ‘default’. The first one is a perfect match to the behavior of `\@makecaption` offered by the standard L<sup>A</sup>T<sub>E</sub>X document classes (and was called ‘default’ in the caption kernel *v1.0*), the second one matches the document class actually used.

```
507 \DeclareCaptionStyle{base}[indent=0pt,justification=centering]{}
508 \DeclareCaptionStyle{default}[indent=0pt,justification=centering]{%
509   format=default,labelsep=default,textformat=default,%
510   justification=default,font=default,labelfont=default,textfont=default}
```

## 8 Formats

```
\DeclareCaptionFormat \DeclareCaptionFormat{<name>}{<code with #1, #2, and #3>}
\DeclareCaptionFormat*{<name>}{<code with #1, #2, and #3>}
```

The starred form causes the code being typeset in vertical (instead of horizontal) mode, but does not support the `indentation=` option.

```

511 \newcommand*\DeclareCaptionFormat{%
512   \caption@teststar\caption@declareformat\@gobble\@firstofone}
513 \@onlypreamble\DeclareCaptionFormat

514 \newcommand*\caption@declareformat[2]{%
515   \@dblarg{\caption@@declareformat#1{#2}}}%
516 \@onlypreamble\caption@declareformat

517 \long\def\caption@@declareformat#1#2[#3]#4{%
518   \global\expandafter\let\csname caption@ifh@#2\endcsname#1%
519   \global\long\@namedef{caption@slfmt@#2}##1##2##3{#3}%
520   \global\long\@namedef{caption@fmt@#2}##1##2##3{#4}}
521 \@onlypreamble\caption@@declareformat

522 \DeclareCaptionOption{format}{\caption@setformat{#1}}

```

`\caption@setformat`    `\caption@setformat{<name>}`

Selecting a caption format simply means saving the code (in `\caption@fmt`) and if the code should be used in horizontal or vertical mode (`\caption@ifh`).

```

523 \newcommand*\caption@setformat[1]{%
524   \ifundefined{caption@fmt@#1}%
525     {\caption@Error{Undefined format `#1'}}%
526     {\expandafter\let\expandafter\caption@ifh\csname caption@ifh@#1\endcsname
527      \expandafter\let\expandafter\caption@slfmt\csname caption@slfmt@#1\endcsname
528      \expandafter\let\expandafter\caption@fmt\csname caption@fmt@#1\endcsname}}

```

`\DeclareCaptionDefaultFormat`

```

529 \newcommand*\DeclareCaptionDefaultFormat[1]{%
530   \expandafter\def\expandafter\caption@fmt@default\expandafter
531     {\csname caption@fmt@#1\endcsname}%
532   \expandafter\def\expandafter\caption@slfmt@default\expandafter
533     {\csname caption@slfmt@#1\endcsname}%
534   \expandafter\def\expandafter\caption@ifh@default\expandafter
535     {\csname caption@ifh@#1\endcsname}}
536 \@onlypreamble\DeclareCaptionDefaultFormat

```

There are two pre-defined formats, called ‘plain’ and ‘hang’.

```

537 \DeclareCaptionFormat{plain}{#1#2#3\par}

538 \DeclareCaptionFormat{hang}[#1#2#3\par]{%
539   \caption@ifin@list\caption@lsepclist\caption@lsepname
540   {\caption@Error{%
541     The option `labelsep=\caption@lsepname' does not work\MessageBreak
542     with `format=hang'}}%
543   {\@hangfrom{#1#2}%
544     \advance\caption@parindent\hangindent
545     \advance\caption@hangindent\hangindent
546     \caption@@par#3\par}}

```

‘default’ usually maps to ‘plain’.

```

547 \DeclareCaptionDefaultFormat{plain}

```

## 9 Label formats

```
DeclareCaptionLabelFormat \DeclareCaptionLabelFormat{<name>}{<code with #1 and #2>}
548 \newcommand*{\DeclareCaptionLabelFormat}[2]{%
549   \global\@namedef{caption@lfmt@#1}##1##2{#2}}
550 \@onlypreamble\DeclareCaptionLabelFormat

551 \DeclareCaptionOption{labelformat}{\caption@setlabelformat{#1}}
```

```
\caption@setlabelformat \caption@setlabelformat{<name>}
```

Selecting a caption label format simply means saving the code (in `\caption@lfmt`).

```
552 \newcommand*{\caption@setlabelformat}[1]{%
553   \ifundefined{caption@lfmt@#1}%
554     {\caption@Error{Undefined label format `#1'}}%
555     {\expandafter\let\expandafter\caption@lfmt\csname caption@lfmt@#1\endcsname}}
```

There are four pre-defined label formats, called ‘empty’, ‘simple’, ‘parens’, and ‘brace’.

```
556 \DeclareCaptionLabelFormat{empty}{}
557 \DeclareCaptionLabelFormat{simple}{\bothIfFirst{#1}{\nobreakspace}#2}
558 \DeclareCaptionLabelFormat{parens}{\bothIfFirst{#1}{\nobreakspace}{#2}}
559 \DeclareCaptionLabelFormat{brace}{\bothIfFirst{#1}{\nobreakspace}#2)}
```

‘default’ usually maps to ‘simple’.

```
560 \def\caption@lfmt@default{\caption@lfmt@simple}
```

## 10 Label separators

```
lareCaptionLabelSeparator \DeclareCaptionLabelSeparator{<name>}{<code>}
\DeclareCaptionLabelSeparator*{<name>}{<code>}
```

The starred form causes the label separator to be typeset *without* using `\captionlabelfont`.

```
561 \newcommand\DeclareCaptionLabelSeparator{%
562   \caption@teststar\caption@declarelabelseparator\@gobble\@firstofone}
563 \@onlypreamble\DeclareCaptionLabelSeparator

564 \newcommand\caption@declarelabelseparator[3]{%
565   \global\@namedef{caption@iflf@#2}{#1}%
566   \global\long\@namedef{caption@lsep@#2}{#3}%
567   \caption@@declarelabelseparator{#2}#3\\@nil}
568 \@onlypreamble\caption@declarelabelseparator

569 \long\def\caption@@declarelabelseparator#1#2\\#3\\@nil{%
570   \def\@tempa{#3}\ifx\@tempa\@empty \else
571     \caption@g@addto@list\caption@lsep@#1}%
572   \fi}
573 \@onlypreamble\caption@@declarelabelseparator

574 \DeclareCaptionOption{labelsep}{\caption@setlabelseparator{#1}}
575 \DeclareCaptionOption{labelseparator}{\caption@setlabelseparator{#1}}
```

```
\caption@setlabelseparator \caption@setlabelseparator{<name>}
```

Selecting a caption label separator simply means saving the code (in `\caption@lsep`).

```
576 \newcommand*{\caption@setlabelseparator}[1]{%
577   \ifundefined{caption@lsep@#1}%
578     {\caption@Error{Undefined label separator `#1'}}%
```

```

579     {\edef\caption@lsepname{#1}%
580     \expandafter\let\expandafter\caption@iflfl\csname caption@iflfl@#1\endcsname
581     \expandafter\let\expandafter\caption@lsep\csname caption@lsep@#1\endcsname}}

```

There are seven pre-defined label separators, called ‘none’, ‘colon’, ‘period’, ‘space’, ‘quad’, ‘newline’, and ‘endash’.

```

582 \DeclareCaptionLabelSeparator{none}{}
583 \DeclareCaptionLabelSeparator{colon}{: }
584 \DeclareCaptionLabelSeparator{period}{. }
585 \DeclareCaptionLabelSeparator{space}{ }
586 \DeclareCaptionLabelSeparator*{quad}{\quad}
587 \DeclareCaptionLabelSeparator*{newline}{\\}
588 \DeclareCaptionLabelSeparator*{endash}{\space\textendash\space}

```

`\caption@setdefaultlabelsep`

```

589 \newcommand*\caption@setdefaultlabelsep[1]{%
590   \ifx\caption@lsep\caption@lsep@default
591     \caption@set@default@labelsep{#1}%
592     \caption@setlabelseparator{default}%
593   \else
594     \caption@set@default@labelsep{#1}%
595   \fi}

596 \newcommand*\caption@set@default@labelsep[1]{%
597   \def\caption@lsep@default{\@nameuse{caption@lsep@#1}}%
598   \def\caption@iflfl@default{\@nameuse{caption@iflfl@#1}}}

```

‘default’ usually maps to ‘colon’.

```

599 \caption@set@default@labelsep{colon}

```

## 11 Text formats

```

\DeclareCaptionTextFormat \DeclareCaptionTextFormat{<name>}{<code with #1>}

600 \newcommand*\DeclareCaptionTextFormat[2]{%
601   \global\long\@namedef{caption@tfmt@#1}##1{#2}}
602 \@onlypreamble\DeclareCaptionTextFormat

603 \DeclareCaptionOption{textformat}{\caption@settextformat{#1}}
604 \DeclareCaptionOption{strut}[1]{\caption@set@bool\caption@ifstrut{#1}}

```

`\caption@settextformat` `\caption@settextformat{<name>}`

Selecting a caption text format simply means saving the code (in `\caption@tfmt`).

```

605 \newcommand*\caption@settextformat[1]{%
606   \@ifundefined{caption@tfmt@#1}%
607   {\caption@Error{Undefined text format `#1'}}%
608   {\expandafter\let\expandafter\caption@tfmt\csname caption@tfmt@#1\endcsname}}

```

There are two pre-defined text formats, called ‘simple’ and ‘period’.

```

609 \DeclareCaptionTextFormat{simple}{#1}
610 \DeclareCaptionTextFormat{period}{#1.}

```

‘default’ usually maps to ‘simple’.

```

611 \def\caption@tfmt@default{\caption@tfmt@simple}

```

## 12 Fonts

```

\DeclareCaptionFont \DeclareCaptionFont{<name>}{<code>}
612 \newcommand*\DeclareCaptionFont[2]{%
613   \define@key{caption@fnt}{#1}[]{\l@addto@macro\caption@fnt{#2}}
614 \@onlypreamble\DeclareCaptionFont

DeclareCaptionDefaultFont \DeclareCaptionDefaultFont{<name>}{<code>}
615 \newcommand*\DeclareCaptionDefaultFont[2]{%
616   \global\@namedef{caption#1@default}{#2}}
617 \@onlypreamble\DeclareCaptionDefaultFont

618 \DeclareCaptionOption{font}{\caption@setfont{font}{#1}}
619 \DeclareCaptionOption{font+}{\caption@addtofont{font}{#1}}
620 \DeclareCaptionDefaultFont{font}{}

621 \DeclareCaptionOption{labelfont}{\caption@setfont{labelfont}{#1}}
622 \DeclareCaptionOption{labelfont+}{\caption@addtofont{labelfont}{#1}}
623 \DeclareCaptionDefaultFont{labelfont}{}

624 \DeclareCaptionOption{textfont}{\caption@setfont{textfont}{#1}}
625 \DeclareCaptionOption{textfont+}{\caption@addtofont{textfont}{#1}}
626 \DeclareCaptionDefaultFont{textfont}{}

\caption@setfont \caption@setfont{<name>}{<keyval-list of names>}
Selecting a caption font means saving all the code snippets in \caption{<name>}.
627 \newcommand*\caption@setfont[1]{%
628   \expandafter\let\csname caption#1\endcsname\@empty
629   \caption@addtofont{#1}}

\caption@addtofont \caption@addtofont{<name>}{<keyval-list of names>}
Like \caption@setfont, but adds the code snippets to \caption{<name>}.
Because we use \setkeys recursive here we need to do this inside an extra group.
630 \newcommand*\caption@addtofont[2]{%
631   \begingroup
632     \expandafter\let\expandafter\caption@fnt\csname caption#1\endcsname
633     \define@key{caption@fnt}{default}[]{%
634       \l@addto@macro\caption@fnt{\csname caption#1@default\endcsname}}%
635     \caption@setkeys[caption]{caption@fnt}{#2}%
636     \global\let\caption@tempa\caption@fnt
637   \endgroup
638   \expandafter\let\csname caption#1\endcsname\caption@tempa}

\caption@font \caption@font{<keyval-list of names>}
\caption@font*{<keyval-code>}
Sets the given font, e.g. \caption@font{small, it} is equivalent to \small\itshape.
639 \newcommand*\caption@font{%
640   \caption@teststar\caption@@font\@firstofone
641   {\caption@setkeys[caption]{caption@fnt}}
642 \newcommand*\caption@@font[2]{%
643   \begingroup
644   \def\caption@fnt{\endgroup}%
645   #1{#2}%
646   \caption@fnt}

```

These are the pre-defined font code snippets.

```

647 \DeclareCaptionFont{normalcolor}{\normalcolor}
648 \DeclareCaptionFont{color}{\color{#1}}

649 \DeclareCaptionFont{normalfont}{\normalfont}
650 \DeclareCaptionFont{up}{\upshape}
651 \DeclareCaptionFont{it}{\itshape}
652 \DeclareCaptionFont{sl}{\slshape}
653 \DeclareCaptionFont{sc}{\scshape}
654 \DeclareCaptionFont{md}{\mdseries}
655 \DeclareCaptionFont{bf}{\bfseries}
656 \DeclareCaptionFont{rm}{\rmfamily}
657 \DeclareCaptionFont{sf}{\sffamily}
658 \DeclareCaptionFont{tt}{\ttfamily}

659 \DeclareCaptionFont{scriptsize}{\scriptsize}
660 \DeclareCaptionFont{footnotesize}{\footnotesize}
661 \DeclareCaptionFont{small}{\small}
662 \DeclareCaptionFont{normalsize}{\normalsize}
663 \DeclareCaptionFont{large}{\large}
664 \DeclareCaptionFont{Large}{\Large}

665 \DeclareCaptionFont{sansmath}{\sansmath}

666 \DeclareCaptionFont{singlespacing}{%
667   \caption@ifundefined{setspace@singlespace}{%
668     \setstretch\setspace@singlespace}% normally 1
669 \DeclareCaptionFont{onehalfspacing}{\onehalfspacing}
670 \DeclareCaptionFont{doublespacing}{\doublespacing}
671 \DeclareCaptionFont{stretch}{\setstretch{#1}}

672 %\DeclareCaptionFont{normal}{%
673 %  \caption@font{normalcolor,normalfont,normalsize,singlespacing}
674 \DeclareCaptionFont{normal}{%
675   \caption@font*{%
676     \KV@caption@fnt@normalcolor\@unused
677     \KV@caption@fnt@normalfont\@unused
678     \KV@caption@fnt@normalsize\@unused
679     \KV@caption@fnt@singlespacing\@unused}}

```

The old versions *v1.x* of the `caption` kernel offered this command to setup the font size used for captions. We still do so old documents will work fine.

```

680 \DeclareCaptionOption{size}{\caption@setfont{size}{#1}}
681 \DeclareCaptionDefaultFont{size}{}

```

## 13 Justifications

```

\DeclareCaptionJustification{\langle name \rangle}{\langle code \rangle}

682 \newcommand*\DeclareCaptionJustification[2]{%
683   \global\@namedef{caption@hj@#1}{#2}% for compatibility to v1.0
684   \DeclareCaptionFont{#1}{#2}}
685 \@onlypreamble\DeclareCaptionJustification

\DeclareCaptionDefaultJustification{\langle code \rangle}

686 \newcommand*\DeclareCaptionDefaultJustification[1]{%

```

```

687 \global\@namedef{caption@hj@default}{#1}% for compatibility to v1.0
688 \DeclareCaptionDefaultFont{@hj}{#1}
689 \@onlypreamble\DeclareCaptionDefaultJustification

690 \DeclareCaptionOption{justification}{\caption@setjustification{#1}}
691 \DeclareCaptionDefaultJustification{}

```

```

\caption@setjustification \caption@setjustification{<name>}
Selecting a caption justification simply means saving the code (in \caption@hj).
692 \newcommand*\caption@setjustification{\caption@setfont{@hj}}

```

These are the pre-defined justification code snippets.

```

693 \DeclareCaptionJustification{justified}{}
694 \DeclareCaptionJustification{centering}{\centering}
695 \DeclareCaptionJustification{centerfirst}{\centerfirst}
696 \DeclareCaptionJustification{centerlast}{\centerlast}
697 \DeclareCaptionJustification{raggedleft}{\raggedleft}
698 \DeclareCaptionJustification{raggedright}{\raggedright}

```

\centerfirst Please blame Frank Mittelbach for the code of \centerfirst :-)

```

699 \providecommand\centerfirst{%
700   \let\\\@centercr
701   \edef\caption@normaladjust{%
702     \leftskip\the\leftskip
703     \rightskip\the\rightskip
704     \parfillskip\the\parfillskip\relax}%
705   \leftskip\z@\@plus -1fil%
706   \rightskip\z@\@plus 1fil%
707   \parfillskip\z@skip
708   \noindent\hskip\z@\@plus 2fil%
709   \@setpar{\@@par\@restorepar\caption@normaladjust}}

```

\centerlast This is based on code from Anne Brüggemann-Klein<sup>[1]</sup>

```

710 \providecommand\centerlast{%
711   \let\\\@centercr
712   \leftskip\z@\@plus 1fil%
713   \rightskip\z@\@plus -1fil%
714   \parfillskip\z@\@plus 2fil\relax}

```

### 13.1 The ragged2e package

We also support the upper-case commands offered by the ragged2e package. Note that these just map to their lower-case variants if the ragged2e package is not available.

```

715 \DeclareCaptionJustification{Centering}{%
716   \caption@ragged\Centering\centering}
717 \DeclareCaptionJustification{RaggedLeft}{%
718   \caption@ragged\RaggedLeft\raggedleft}
719 \DeclareCaptionJustification{RaggedRight}{%
720   \caption@ragged\RaggedRight\raggedright}

```

\caption@ragged \caption@ragged will be basically defined as



```

\AtBeginDocument{\IfFileExists{ragged2e.sty}%
  {\RequirePackage{ragged2e}\let\caption@ragged\@firstoftwo}%
  {\let\caption@ragged\@secondoftwo}}

```

but with an additional warning if the ragged2e package is not loaded (yet). (This warning will be type out only one time per option, that's why we need the `\caption\string#1` stuff.) Furthermore we load the ragged2e package, if needed and available.

```

721 \newcommand*\caption@ragged{%
722   \caption@Debug{We need ragged2e}%
723   \protected@write\@auxout{}\string\caption@newlabel{ragged2e}{}%
724   \global\let\caption@ragged\caption@@ragged
725   \caption@ragged}

726 \caption@AtBeginDocument{%
727   \@ifundefined{caption@r@ragged2e}{%
728     \newcommand*\caption@@ragged{%
729       \caption@Warning{%
730         'ragged2e' support has been changed.\MessageBreak
731         Rerun to get captions right}%
732       \global\let\caption@ragged\@secondoftwo % suppress further warnings
733       \caption@ragged}%
734   }{%
735     \caption@Debug{We load ragged2e}%
736     \IfFileExists{ragged2e.sty}{%
737       \RequirePackage{ragged2e}%
738       \let\caption@@ragged\@firstoftwo
739     }{%
740       \newcommand*\caption@@ragged[2]{%
741         \@ifundefined{caption\string#1}{%
742           \caption@Warning{%
743             'ragged2e' package not loaded, therefore\MessageBreak
744             substituting \string#2 for \string#1\MessageBreak}%
745           \global\@namedef{caption\string#1}{}%
746           #2}%
747         }%
748       }%

```

## 14 Vertical spaces before and after captions

`\abovecaptionskip` Usually these skips are defined within the document class, but some document classes don't do so.

```

\belowcaptionskip
749 \caption@ifundefined\abovecaptionskip{%
750   \newlength\abovecaptionskip\setlength\abovecaptionskip{10\p@}{}
751 \caption@ifundefined\belowcaptionskip{%
752   \newlength\belowcaptionskip\setlength\belowcaptionskip{0\p@}{}

753 \DeclareCaptionOption{aboveskip}{\setlength\abovecaptionskip{#1}}
754 \DeclareCaptionOption{belowskip}{\setlength\belowcaptionskip{#1}}
755 \DeclareCaptionOption{skip}{\setlength\abovecaptionskip{#1}}

```

`\caption@rule` `\caption@rule`

Draws an invisible rule to adjust the “skip” setting.

```

756 \newcommand*\caption@rule{\caption@ifrule\caption@hrule\relax}

```

```

757 \newcommand*\caption@hrule{\hrule\@height\z@}
758 \DeclareCaptionOption{rule}[1]{\caption@set@bool\caption@ifrule{#1}}

```

## 15 Positioning

These macros handle the right position of the caption. Note that the position is actually *not* controlled by the `caption3` kernel options, but by the user (or a specific package like the `float` package) instead. The user can put the `\caption` command wherever he likes! So this stuff is only to give us a *hint* where to put the right skips, the user usually has to take care for himself that this hint actually matches the right position.

```

759 \DeclareCaptionOption{position}{\caption@setposition{#1}}

```

`\caption@setposition`    `\caption@setposition{<position>}`

Selecting the caption position means that we put `\caption@position` to the right value. *Please do **not** use the internal macro `\caption@position` in your own package or document, but use the wrapper macro `\caption@iftop` instead.*

```

760 \newcommand*\caption@setposition[1]{%
761   \caption@ifinlist{#1}{d,default}{%
762     \let\caption@position\caption@defaultpos
763   }\caption@ifinlist{#1}{t,top,above}{%
764     \let\caption@position\@firstoftwo
765   }\caption@ifinlist{#1}{b,bottom,below}{%
766     \let\caption@position\@secondoftwo
767   }\caption@ifinlist{#1}{a,auto}{%
768     \let\caption@position\@undefined
769   }{%
770     \caption@Error{Undefined position `#1'}%
771   }}}

```

`\caption@defaultpos`

The default ‘position’ is ‘auto’, this means that the `caption` kernel will try to guess the current position of the caption. (But in many cases, for example in `longtables`, this is doomed to fail!)

The setting ‘bottom’ corresponds to the `\@makecaption` implementation in the standard `LATEX` document classes, but ‘auto’ should give better results in most cases.

```

772 %\caption@setdefaultpos{a}% default = auto
773 \let\caption@defaultpos\@undefined

```

`\caption@iftop`

`\caption@iftop{<true-code>}{<false-code>}`  
(If the `position=` is set to `auto` we assume a bottom position here.)

```

774 \newcommand*\caption@iftop{%
775   \ifx\caption@position\@undefined
776     \let\caption@position\@secondoftwo
777 %   = \caption@setposition b%
778   \fi
779   \caption@position}

```

`\caption@fixposition`

`\caption@fixposition`

This macro checks if the ‘position’ is set to ‘auto’. If yes, `\caption@autoposition` will be called to set `\caption@position` to a proper value we can actually use.

```

780 \newcommand*\caption@fixposition{%

```

```

781 \ifx\caption@position\@undefined
782 \caption@autoposition
783 \fi}

```

`\caption@autoposition` `\caption@autoposition`

We guess the current position of the caption by checking `\prevdepth`.

A different solution would be setting the `\spacefactor` to something not much less than 1000 (for example 994) in `\caption@start` and checking this value here by `\ifnum\spacefactor=994`. (It's implemented in the `threeparttable` package<sup>[4]</sup> this way.)

Another idea would be checking `\@ifminipage`, but since some packages typeset the caption within a simple `\vbox` this does not seem to be a good one.

```

784 \newcommand*\caption@autoposition{%
785 \ifvmode
786 \edef\caption@tempa{\the\prevdepth}%
787 \caption@Debug{\protect\prevdepth=\caption@tempa}%
788 \ifdim\prevdepth>-\p@
789 \let\caption@position\@secondoftwo
790 \else
791 \let\caption@position\@firstoftwo
792 \fi
793 % = \caption@setposition{\ifdim\prevdepth>-\p@ b\else t\fi}%
794 \else
795 \caption@Debug{no \protect\prevdepth}%
796 \let\caption@position\@secondoftwo
797 % = \caption@setposition b%
798 \fi}

```

`\caption@setautoposition` `\caption@setautoposition{<position>}`

replaces the above algorithm by a different one (or a fixed position setting).

```

799 \newcommand*\caption@setautoposition[1]{%
800 \def\caption@autoposition{\caption@setposition{#1}}

```

## 16 Hooks

`\AtBeginCaption` `\AtBeginCaption {<code>}`  
`\AtEndCaption` `\AtEndCaption {<code>}`

These hooks can be used analogous to `\AtBeginDocument` and `\AtEndDocument`.

```

801 \newcommand*\caption@beginhook{}
802 \newcommand*\caption@endhook{}
803 \newcommand*\AtBeginCaption{\l@addto@macro\caption@beginhook}
804 \newcommand*\AtEndCaption{\l@addto@macro\caption@endhook}

```

## 17 Lists

```

805 \DeclareCaptionOption{list}[1]{\caption@setlist{#1}}
806 \DeclareCaptionOption{listof}[1]{\caption@setlist{#1}}

\caption@setlist \caption@setlist{<boolean>}
807 \newcommand*\caption@setlist{\caption@set@bool\caption@iflist}

```

```

\DeclareCaptionListFormat \DeclareCaptionListFormat{<name>}{<code with #1 and #2>}
808 \newcommand*\DeclareCaptionListFormat[2]{%
809   \global\@namedef{caption@lstfmt@#1}##1##2{#2}}
810 \@onlypreamble\DeclareCaptionListFormat

811 \DeclareCaptionOption{listformat}{\caption@setlistformat{#1}}

\caption@setlistformat \caption@setlistformat{<name>}
Selecting a caption list format simply means saving the code (in \caption@lstfmt).
812 \newcommand*\caption@setlistformat[1]{%
813   \ifundefined{caption@lstfmt@#1}%
814     {\caption@Error{Undefined list format `#1'}}%
815     {\expandafter\let\expandafter\caption@lstfmt
816       \csname caption@lstfmt@#1\endcsname}}

```

There are five pre-defined list formats, taken from the subfig package.

```

817 \DeclareCaptionListFormat{empty}{}
818 \DeclareCaptionListFormat{simple}{#1#2}
819 \DeclareCaptionListFormat{parens}{#1(#2)}
820 \DeclareCaptionListFormat{subsimple}{#2}
821 \DeclareCaptionListFormat{subparens}{(#2)}

tion@setdefaultlistformat
822 \newcommand*\caption@setdefaultlistformat[1]{%
823   \ifx\caption@lstfmt\caption@lstfmt@default
824     \caption@set@default@listformat{#1}%
825     \caption@setlistformat{default}%
826   \else
827     \caption@set@default@listformat{#1}%
828   \fi}

829 \newcommand*\caption@set@default@listformat[1]{%
830   \def\caption@lstfmt@default{\@nameuse{caption@lstfmt@#1}}}

```

‘default’ usually maps to ‘subsimple’.

```
831 \caption@set@default@listformat{subsimple}
```

## 18 Debug option

```

832 \DeclareCaptionOption{debug}[1]{%
833   \caption@set@bool\caption@ifdebug{#1}%
834   \caption@ifdebug
835     {\let\caption@Debug\caption@Info}%
836     {\let\caption@Debug@gobble}}

837 \DeclareOption{debug}{\setkeys{caption}{debug}}
838 \setkeys{caption}{debug=0}

```

## 19 Document classes & Babel support

### 19.1 The standard L<sup>A</sup>T<sub>E</sub>X classes

```
839 \caption@CheckCommand\@makecaption{%

```

```

840 % article|report|book [2005/09/16 v1.4f Standard LaTeX document class]
841 \long\def\@makecaption#1#2{%
842   \vskip\abovecaptionskip
843   \sbox\@tempboxa{#1: #2}%
844   \ifdim \wd\@tempboxa >\hsize
845     #1: #2\par
846   \else
847     \global \@minipagefalse
848     \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
849   \fi
850   \vskip\belowcaptionskip}}

```

## 19.2 The $\mathcal{AMS}$ & SMF classes

\caption@ifamsclass

```

851 \providecommand*\caption@ifamsclass{%
852   \caption@ifundefined\@captionheadfont\@gobble\@firstofone}
853 \@onlypreamble\caption@ifamsclass

854 \caption@ifamsclass{%
855   \caption@CheckCommand\@makecaption{%
856     % amsart|amsproc|amsbook [2004/08/06 v2.20]
857     \long\def\@makecaption#1#2{%
858       \setbox\@tempboxa\vbox{\color@setgroup
859         \advance\hsize-2\captionindent\noindent
860         \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
861           {\@cdr#2\@nil}\{.\@captionfont\upshape\enspace#2}%
862         \unskip\kern-2\captionindent\par
863         \global\setbox\@ne\lastbox\color@endgroup}%
864       \ifhbox\@ne % the normal case
865         \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
866       \fi
867       \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
868         \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
869       \else % tempboxa contained more than one line
870         \setbox\@ne\vbox{\unvbox\@tempboxa\parskip\z@skip
871           \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
872       \fi
873       \ifnum\@tempcnta<64 % if the float IS a figure...
874         \addvspace\abovecaptionskip
875         \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
876       \else % if the float IS NOT a figure...
877         \hbox to\hsize{\kern\captionindent\box\@ne\hss}%
878         \nobreak
879         \vskip\belowcaptionskip
880       \fi
881       \relax
882     }}

883 \caption@CheckCommand\@makecaption{%
884   % smfart|smfbook [1999/11/15 v1.2f Classe LaTeX pour les articles publiés par
885   \long\def\@makecaption#1#2{%
886     \ifdim\captionindent>.1\hsize \captionindent.1\hsize \fi
887     \setbox\@tempboxa\vbox{\color@setgroup
888       \advance\hsize-2\captionindent\noindent

```

```

889      \@captionfont\@captionheadfont#1\@xp\@ifnotempty\@xp
890      {\@cdr#2\@nil}\@addpunct{.}\@captionfont\upshape\enspace#2}%
891      \unskip\kern-2\captionindent\par
892      \global\setbox\@ne\lastbox\color@endgroup}%
893      \ifhbox\@ne % the normal case
894      \setbox\@ne\hbox{\unhbox\@ne\unskip\unskip\unpenalty\unkern}%
895      \fi
896      \ifdim\wd\@tempboxa=\z@ % this means caption will fit on one line
897      \setbox\@ne\hbox to\columnwidth{\hss\kern-2\captionindent\box\@ne\hss}%
898      \@tempdima\wd\@ne\advance\@tempdima-\captionindent
899      \wd\@ne\@tempdima
900      \else % tempboxa contained more than one line
901      \setbox\@ne\vbox{\rightskip=0pt plus\captionindent\relax
902      \unvbox\@tempboxa\parskip\z@skip
903      \noindent\unhbox\@ne\advance\hsize-2\captionindent\par}%
904      \fi
905      \ifnum\@tempcnta<64 % if the float IS a figure...
906      \addvspace\abovecaptionskip
907      \noindent\kern\captionindent\box\@ne
908      \else % if the float IS NOT a figure...
909      \noindent\kern\captionindent\box\@ne
910      \nobreak
911      \vskip\belowcaptionskip
912      \fi
913      \relax
914      }}
915      \let\captionmargin\captionindent % set to 3pc by AMS class
916      \begingroup\edef\@tempa{\endgroup
917      \noexpand\caption@g@addto@list\noexpand\caption@sty@default
918      {margin=\the\captionmargin
919      \caption@ifundefined\smf@makecaption{{},maxmargin=.1\linewidth}}}}
920      \@tempa
921      \caption@g@addto@list\caption@sls@default{margin*=.5\captionmargin}
922      \DeclareCaptionLabelSeparator{default}{.\enspace}
923      \DeclareCaptionDefaultFont{font}{\@captionfont}
924      \DeclareCaptionDefaultFont{labelfont}{\@captionheadfont}
925      \DeclareCaptionDefaultFont{textfont}{\@captionfont\upshape}
926      \captionsetup[figure]{position=b}
927      \captionsetup[table]{position=t}
928 }

```

### 19.3 The beamer class (Part one)

\caption@ifbeamerclass

```

929 \providecommand*\caption@ifbeamerclass{%
930   \@ifclassloaded{beamer}\@firstofone\@gobble}
931 \onlypreamble\caption@ifbeamerclass

932 \caption@ifbeamerclass{%
933   \caption@CheckCommand\beamer@makecaption{%
934     % beamerbaselocalstructure.sty,v 1.53 2007/01/28 20:48:21 tantau
935     \long\def\beamer@makecaption#1#2{%
936       \def\insertcaptionname{\csname#1name\endcsname}%

```

```

937 \def\insertcaptionnumber{\csname the#1\endcsname}%
938 \def\insertcaption{#2}%
939 \nobreak\vskip\abovcaptionskip\nobreak
940 \sbox\@tempboxa{\usebeamertemplate**{caption}}%
941 \ifdim \wd\@tempboxa >\hsize
942 \usebeamertemplate**{caption}\par
943 \else
944 \global \@minipagefalse
945 \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
946 \fi
947 \nobreak\vskip\belowcaptionskip\nobreak}}

```

\caption@ifbeamertemplate

```

948 \newcommand*\caption@ifbeamertemplate[1]{%
949 \begingroup
950 \let\beamer@@tpl\caption@ORI\beamer@@tpl\caption
951 \@nameuse{beamer@@tmpop\caption@#1}%
952 \ifx\beamer@@tpl\caption@ORI\beamer@@tpl\caption
953 \endgroup\expandafter\@firstoftwo
954 \else
955 \endgroup\expandafter\@secondoftwo
956 \fi}

957 \DeclareCaptionLabelFormat{default}{%
958 #1\caption@ifbeamertemplate{numbered}{~#2}{}}
959 \caption@declarelabelseparator
960 {\caption@ifbeamertemplate{caption name own line}\@gobble\@firstofone}
961 {default}
962 {\caption@ifbeamertemplate{caption name own line}{\}\{: }}
963 \DeclareCaptionDefaultFont{font}{%
964 \usebeamerfont*{caption}%
965 \usebeamerfont*{caption}%
966 \DeclareCaptionDefaultFont{labelfont}{%
967 \usebeamerfont*{caption name}%
968 \usebeamerfont*{caption name}}
969 \DeclareCaptionDefaultJustification{\raggedright}
970 \DeclareOption{beamerclass}{%
971 \renewcommand*\caption@ifslc{%
972 \caption@ifbeamertemplate{caption name own line}\@secondoftwo\@firstoftwo}
973 % Since the beamer class do not offer a 'list of figures' we switch this supp
974 \captionsetup{list=0}}
975 \PassOptionsToPackage{beamerclass}{caption3}

```

If the beamer document class is used, we offer a beamer template called ‘caption3’ which can be used with option ‘beamer’ or \setbeamertemplate{caption}[caption3]. (Note that this is of no use when the caption package is used, too.)

```

976 \defbeamertemplate{caption}{caption3}{%
977 \caption@make\insertcaptionname\insertcaptionnumber\insertcaption}
978 \DeclareOption{beamer}{%
979 % \usebeamertemplate**{caption} will set font
980 \DeclareCaptionDefaultFont{font}{}%
981 \setbeamertemplate{caption}[caption3]}
982 %

```

```

983 %      \begin{macrocode]
984 }

```

## 19.4 The KOMA-Script classes

\caption@ifkomaaclass

```

985 \providecommand*\caption@ifkomaaclass{%
986   \caption@ifundefined\scr@caption\@gobble\@firstofone}
987 \onlypreamble\caption@ifkomaaclass

988 \caption@ifkomaaclass{%
989   \caption@CheckCommand\@makecaption{%
990     % scrartcl|scrreprt|scrbook [2007/03/07 v2.97a KOMA-Script document class]
991     \long\def\@makecaption#1#2{%
992       \if@captionabove
993         \vskip\belowcaptionskip
994       \else
995         \vskip\abovecaptionskip
996       \fi
997       \@@makecaption\@firstofone{#1}{#2}%
998       \if@captionabove
999         \vskip\abovecaptionskip
1000      \else
1001        \vskip\belowcaptionskip
1002      \fi}}
1003 \DeclareCaptionFormat{default}[#1#2#3\par]{%
1004   \ifdofullc@p
1005     \caption@ifin@list\caption@lsep@crlist\caption@lsepname
1006     {\caption@Error{%
1007       The option 'labelsep=\caption@lsepname' does not work\MessageBreak
1008       with \noexpand\setcaphanging (which is set by default)}}%
1009     {\caption@fmt@hang{#1}{#2}{#3}}%
1010   \else
1011     #1#2%
1012     \ifdim\cap@indent<\z@
1013       \par
1014       \noindent\hspace*{-\cap@indent}%
1015     \else\if@capbreak
1016       \par
1017     \fi\fi
1018     #3\par
1019   \fi}
1020 \DeclareCaptionLabelSeparator{default}{\captionformat}
1021 \DeclareCaptionDefaultFont{font}{\scr@fnt@caption}
1022 \DeclareCaptionDefaultFont{labelfont}{\scr@fnt@captionlabel}
1023 }

```

## 19.5 The NTG Dutch classes

\caption@ifntgclass

```

1024 \providecommand*\caption@ifntgclass{%
1025   \caption@ifundefined\CaptionFonts\@gobble\@firstofone}
1026 \onlypreamble\caption@ifntgclass

```



```

1027 \caption@ifntgclass{%
1028   \caption@CheckCommand\@makecaption{%
1029     % artikel|rapport|boek [2004/06/07 v2.1a NTG LaTeX document class]
1030     \long\def\@makecaption#1#2{%
1031       \vskip\abovecaptionskip
1032       \sbox\@tempboxa{\CaptionLabelFont#1:} \CaptionTextFont#2}%
1033       \ifdim \wd\@tempboxa >\hsize
1034         {\CaptionLabelFont#1:} \CaptionTextFont#2\par
1035       \else
1036         \global \@minipagefalse
1037         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1038       \fi
1039       \vskip\belowcaptionskip}}
1040 \DeclareCaptionDefaultFont{labelfont}{\CaptionLabelFont}
1041 \DeclareCaptionDefaultFont{textfont}{\CaptionTextFont}
1042 }

```

## 19.6 The thesis class

\caption@ifthesisclass

```

1043 \providecommand*\caption@ifthesisclass{%
1044   \caption@ifundefined\cph@font{\@gobble}{\caption@ifundefined\cpb@font{\@gobble}\@
1045 \caption@ifthesisclass{%
1046   \caption@CheckCommand\@makecaption{%
1047     % thesis.cls 1996/25/01 1.0g LaTeX document class (wm).
1048     \long\def\@makecaption#1#2{%
1049       \vskip\abovecaptionskip
1050       \setbox\@tempboxa\hbox{{\cph@font #1:} {\cpb@font #2}}%
1051       \ifdim \wd\@tempboxa >\hsize
1052         \@hangfrom{\cph@font #1:} {\cpb@font #2\par}%
1053       \else
1054         \hbox to\hsize{\hfil\box\@tempboxa\hfil}%
1055       \fi
1056       \vskip\belowcaptionskip}}
1057 \DeclareCaptionDefaultFormat{hang}
1058 \DeclareCaptionDefaultFont{labelfont}{\cph@font}
1059 \DeclareCaptionDefaultFont{textfont}{\cpb@font}
1060 }

```

## 19.7 The frenchb Babel option

```

1061 \caption@ifundefined\FB@makecaption{}{%
1062   \caption@CheckCommand\@makecaption{%
1063     % frenchb.ldf [2005/02/06 v1.6g French support from the babel system]
1064     % frenchb.ldf [2007/10/05 v2.0e French support from the babel system]
1065     \long\def\@makecaption#1#2{%
1066       \vskip\abovecaptionskip
1067       \sbox\@tempboxa{#1\CaptionSeparator #2}%
1068       \ifdim \wd\@tempboxa >\hsize
1069         #1\CaptionSeparator #2\par

```

```

1070     \else
1071       \global \@minipagefalse
1072       \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1073     \fi
1074     \vskip\belowcaptionskip}}
1075 \ifx\@makecaption\STD@makecaption
1076   \DeclareCaptionLabelSeparator{default}{\CaptionSeparator}
1077   \def\caption@frenchb{% supress frenchb warning
1078     \let\STD@makecaption\@makecaption
1079     \let\FB@makecaption\@makecaption}
1080 \else
1081   \ifx\@makecaption\@undefined\else
1082     \caption@InfoNoLine{%
1083       The definition of \protect\@makecaption\space
1084       has been changed,\MessageBreak
1085       frenchb will NOT customize it}%
1086   \fi
1087 \fi
1088 }

```

## 19.8 The frenchle/pro package

```

1089 \caption@ifundefined\frenchTeXmods{}{%
1090   \caption@CheckCommand\@makecaption{%
1091     % french(1e).sty [2006/10/03 The french(1e) package /V5,9991/]
1092     % french(1e).sty [2007/06/28 The french(1e) package /V5,9994/]
1093     \def\@makecaption#1#2{%
1094       \ifFTY%
1095         \def\@secondofmany##1##2\void{##2}%
1096         \def\@tempa{\@secondofmany#2\void}%
1097         \ifx\@tempa\empty%
1098           \let\captionseparator\empty%
1099         \fi%
1100         \@mcORI{#1}{\relax\captionfont{#2}}%
1101       \else
1102         \@mcORI{#1}{#2}%
1103       \fi}}%
1104   \caption@CheckCommand\@makecaption{%
1105     % french(1e).sty [2007/02/11 The french(1e) package /V5,9993/]
1106     \def\@makecaption#1#2{%
1107       \ifFTY%
1108         \def\@secondofmany##1##2\void{##2}%
1109         \protected@edef\@tempa{\@secondofmany#2\void}%
1110         \ifx\@tempa\empty%
1111           \let\captionseparator\empty%
1112         \fi%
1113         \@mcORI{#1}{\relax\captionfont{#2}}%
1114       \else
1115         \@mcORI{#1}{#2}%
1116       \fi}}%
1117   \DeclareCaptionDefaultFont{textfont}{\itshape}%
1118   \DeclareCaptionLabelSeparator{default}{\captionseparator\space}%

```

```
1119 }
```

## 19.9 The hungarian and magyar Babel option

```
1120 \DeclareCaptionListFormat{subperiod}{#2.}
1121 \caption@ifundefined{hunnewlabel}{}{%
1122   \caption@CheckCommand\@makecaption{%
1123     % magyar.ldf [2005/03/30 v1.4j Magyar support from the babel system]
1124     \def\@makecaption#1#2{%
1125       \vskip\abovecaptionskip
1126       \sbox\@tempboxa{#1. #2}%
1127       \ifdim \wd\@tempboxa >\hsize
1128         {#1. #2\csname par\endcsname}
1129       \else
1130         \global \@minipagefalse
1131         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
1132       \fi
1133       \vskip\belowcaptionskip}}%
1134 \def\caption@tempa#1{\@ifundefined{extras#1}{}{%
1135   \expandafter\addto\csname extras#1\endcsname{%
1136     % change default labelsep and listformat
1137     \caption@setdefaultlabelsep{period}%
1138     \caption@setdefaultlistformat{subperiod}}%
1139   \expandafter\addto\csname noextras#1\endcsname{%
1140     % change default labelsep and listformat
1141     \caption@setdefaultlabelsep{colon}%
1142     \caption@setdefaultlistformat{subsimple}}%
1143 }}
1144 \caption@tempa{hungarian}
1145 \caption@tempa{magyar}
```

## 19.10 Unknown document class (or package)

```
1146 \caption@ifCheckCommand{%
1147   \caption@setbool{documentclass}{1}%
1148 }{%
1149   \caption@setbool{documentclass}{0}%
1150   \caption@InfoNoLine{%
1151     Unknown document class (or package), \MessageBreak
1152     standard defaults will be used}%
1153   \caption@Debug{\string\@makecaption\space=\space\meaning\@makecaption\@gobble}%
1154 }
```

## 20 Execution of options

```
1155 \captionsetup{style=default,position=default,%
1156               list,listformat=default,twoside=\if@twoside 1\else 0\fi}
1157 \ProcessOptions*
```

## 21 Making an ‘List of’ entry

```
\caption@addcontentsline \caption@addcontentsline{<type>}{<list entry>}
```

Makes an entry in the list-of-whatever, if requested, i.e. the argument *<list entry>* is not empty and `listof=` was set to `true`.

```

1158 \newcommand*\caption@addcontentsline[2]{%
1159   \caption@iflist
1160     {\def\@tempa{#2}}%
1161     {\let\@tempa\@empty}%
1162   \ifx\@tempa\@empty \else
1163     {\let\\space
1164      \ifundefined{ext@#1}%
1165        {\caption@Error{No float type '#1' defined}}%
1166        {\caption@@addcontentsline
1167          {\csname ext@#1\endcsname}%
1168          {#1}%
1169          {\caption@lstfmt{\@nameuse{p@#1}}{\@nameuse{the#1}}}%
1170          {\ignorespaces #2}}}%
1171   \fi}

1172 \newcommand*\caption@@addcontentsline[4]{%
1173   \addcontentsline{#1}{#2}{\protect\numberline{#3}{#4}}}

```

## 22 Typesetting the caption

```

\ifcaption@star  If the starred form of \caption is used, this will be set to true. (It will be reset to
                  false at the end of \caption@@make.)
1174 \newif\ifcaption@star

\caption@fnum    \caption@fnum{<float type>}
                  Typesets the caption label; as replacement for \fnum{<float type>}.
1175 \newcommand*\caption@fnum[1]{\caption@lfmt{\@nameuse{#1name}}{\@nameuse{the#1}}}

\caption@make    \caption@make{<float name>}{<ref. number>}{<text>}
                  Typesets the caption.
1176 \newcommand\caption@make[2]{\caption@@make{\caption@lfmt{#1}{#2}}}

\caption@@make    \caption@@make{<caption label>}{<caption text>}
1177 \newcommand\caption@@make[2]{%
1178   \begingroup
1179   \caption@stepcounter
1180   \caption@beginhook

Check margin, if \caption@minmargin or \caption@maxmargin is set
1181 % TODO: Move this to \caption@calcmargin!?
1182 \ifx\caption@maxmargin\undefined \else
1183   \ifdim\captionmargin>\caption@maxmargin\relax
1184     \captionmargin\caption@maxmargin\relax
1185   \fi
1186 \fi
1187 \ifx\caption@minmargin\undefined \else
1188   \ifdim\captionmargin<\caption@minmargin\relax
1189     \captionmargin\caption@minmargin\relax
1190   \fi
1191 \fi

```

Special single-line treatment (option singlelinecheck=)

```
1192 \caption@ifslc{\caption@slc{#1}{#2}\captionwidth\relax}{}%
```

Typeset the left margin (option margin=)

```
1193 \caption@calcmargin
1194 \@tempdima\captionmargin
1195 \ifdim\captionmargin@=\z@ \else
1196   \caption@ifoddpage{}{\advance\@tempdima\captionmargin@}%
1197 \fi
1198 \caption@ifh{\advance\@tempdima\caption@indent}%
1199 \hspace\@tempdima
```

We actually use a `\vbox` of width `\captionwidth - \caption@indent` to typeset the caption.

*Note:* `\captionindent` is *not* supported if the caption format was defined with `\DeclareCaptionFormat*`.

```
1200 \@tempdima\captionwidth
1201 \caption@ifh{\advance\@tempdima-\caption@indent}%
1202 \caption@parbox\@tempdima{%
```

Typeset the indentation (option indentation=)

Bugfix 04-05-05: `\hskip-\caption@indent` replaced by `\ifdim\caption@indent=\z@...`

```
1203 \caption@ifh{%
1204   \ifdim\caption@indent=\z@
1205     \leavevmode
1206   \else
1207     \hskip-\caption@indent
1208   \fi}%
```

Typeset the caption itself and close the `\caption@parbox`

```
1209 \caption@@@make{#1}{#2}}%
```

Typeset the right margin (option margin=)

```
1210 \@tempdima\captionmargin
1211 \ifdim\captionmargin@=\z@ \else
1212   \caption@ifoddpage{\advance\@tempdima\captionmargin@}{}%
1213 \fi
1214 \hspace\@tempdima
1215 \caption@endhook
1216 \endgroup
1217 \global\caption@starfalse}
```

```
\caption@calcmargin \caption@calcmargin
```

Calculate `\captionmargin` & `\captionwidth`, so both contain valid values.

```
1218 \newcommand*\caption@calcmargin{%
1219 \caption@calcmargin@hook
1220 \ifdim\captionwidth=\z@
1221   \captionwidth\linewidth
1222   \advance\captionwidth by -2\captionmargin
1223   \advance\captionwidth by -\captionmargin@
1224 \else
1225   \captionmargin\linewidth
1226   \advance\captionmargin by -\captionwidth
1227   \divide\captionmargin by 2
1228   \captionmargin@\z@
1229 \fi
```

```

1230 \caption@Debug{%
1231   \string\hsize=\the\hsize,
1232   \string\linewidth=\the\linewidth,\MessageBreak
1233   \string\leftmargin=\the\leftmargin,
1234   \string\rightmargin=\the\rightmargin,\MessageBreak
1235   \string\margin=\the\captionmargin,
1236   \string\margin@=\the\captionmargin@,
1237   \string\width=\the\captionwidth}%
1238 }

\caption@slc \caption@slc{<label>}{<text>}{<width>}{<extra code>}
This one does the single-line-check.
1239 \newcommand\caption@slc[4]{%
1240   \caption@@slc{#1}{#2}{#3}{\caption@singleline#4}{}}
1241 \newcommand\caption@@slc[5]{%
1242   \caption@Debug{Begin SLC}%
1243   \begingroup
1244   \caption@singleline
1245   \let\caption@hj@\empty
1246   \caption@calcmargin % calculate #3 if necessary
1247   \caption@prepareslc
1248   \sbox\@tempboxa{\caption@@make{#1}{#2}}%
1249   \ifdim\wd\@tempboxa>#3%
1250     \endgroup
1251     #5%
1252   \else
1253     \endgroup
1254     #4%
1255   \fi
1256   \caption@Debug{End SLC}}
1257 \newcommand*\caption@singleline{%
1258   \caption@xsetup\caption@opt@singleline
1259   \let\caption@fmt\caption@slfmt}

\caption@prepareslc \caption@prepareslc
Re-define anything which would disturb the single-line-check.
1260 \newcommand*\caption@prepareslc{%
1261   \def\label{\caption@withoptargs\@gobbletwo}%
1262   \let\caption@footnote@ORI\footnote
1263   \def\footnote{\caption@withoptargs\caption@footnote}%
1264   \let\caption@footnotemark@ORI\footnotemark
1265   \def\footnotemark{\caption@withoptargs\caption@footnotemark}%
1266   \let\@footnotetext\@gobble
1267   \let\@endnotetext\@gobble}
1268 \newcommand\caption@footnote[2]{%
1269   \begingroup
1270     \let\stepcounter\caption@l@stepcounter
1271     \caption@footnote@ORI#1{#2}%
1272   \endgroup}
1273 \newcommand\caption@footnotemark[1]{%
1274   \begingroup

```

```

1275 \let\stepcounter\caption@l@stepcounter
1276 \caption@footnotemark@ORI#1%
1277 \endgroup}

1278 \newcommand*\caption@l@stepcounter[1]{%
1279 \advance\csname c@#1\endcsname\@ne\relax}

\caption@parbox \caption@parbox{<width>}{<contents>}
This macro defines the box which surrounds the caption paragraph.
1280 \newcommand*\caption@parbox{\parbox[b]}

\caption@applyfont \caption@applyfont
This macro executes the font relevant macros, i.e. by default the options set by
justification=, font=, and size=.
1281 \newcommand*\caption@applyfont{%
1282 \caption@hj\captionfont\captionsize}

\caption@@@make \caption@@@make{<caption label>}{<caption text>}
This one finally typesets the caption paragraph, without margin and indentation.
1283 \newcommand\caption@@@make[2]{%
If the label is empty, we use no caption label separator.
1284 \sbox\@tempboxa{#1}%
1285 \ifdim\wd\@tempboxa=\z@
1286 \let\caption@lsep\relax
1287 % \@capbreakfalse
1288 \fi

If the text is empty, we use no caption label separator, too. (And no text format either.)
1289 \caption@ifempty{#2}{%
1290 \let\caption@lsep\@empty
1291 \let\caption@tfmt\@firstofone
1292 % \@capbreakfalse
1293 \let\caption@ifstrut\@secondoftwo
1294 }%

Take care that \caption@parindent and \caption@hangindent will be used
to typeset the paragraph.
1295 \@setpar{\@par\caption@par}\caption@@par

Finally typeset the caption.
1296 \caption@applyfont
1297 \caption@fmt
1298 {\ifcaption@star\else\caption@labelfont#1\fi}%
1299 {\ifcaption@star\else\caption@iflf\caption@labelfont\caption@lsep\fi}%
1300 {\caption@textfont
1301 \caption@ifstrut{\vrule\@height\ht\strutbox\@width\z@}{}}%
1302 \nobreak\hskip\z@skip % enable hyphenation
1303 \caption@tfmt{#2}%
1304 % \caption@ifstrut{\vrule\@height\z@\@depth\dp\strutbox\@width\z@}{}}%
1305 \caption@ifstrut{\ifhmode\@finalstrut\strutbox\fi}{}}%
1306 \par}}

\caption@ifempty \caption@ifempty{<text>}{<true>}(no <false>))
This one tests if the <text> is actually empty.

```

*Note:* This will be done without expanding the text, therefore this is far away from being bullet-proof.

*Note:* This macro is re-defining itself so only the first test (in a group) will actually be done.

```

1307 \newcommand\caption@ifempty[1]{%
1308   \caption@if@empty{#1}%
1309   \caption@ifempty\@unused}

1310 \newcommand\caption@if@empty[1]{%
1311   \def\caption@tempa{#1}%
1312   \ifx\caption@tempa\@empty
1313     \let\caption@ifempty\@secondoftwo
1314   \else
1315     \expandafter\def\expandafter\caption@tempa\expandafter{%
1316       \caption@car#1\caption@if@empty\caption@nil}%
1317     \def\caption@tempb{\caption@if@empty}%
1318     \ifx\caption@tempa\caption@tempb
1319       \let\caption@ifempty\@secondoftwo
1320     \else
1321       \def\caption@tempb{\ignorespaces}%
1322       \ifx\caption@tempa\caption@tempb
1323         \expandafter\caption@if@empty\expandafter{\@gobble#1}%
1324       \else
1325         \def\caption@tempb{\label}%
1326         \ifx\caption@tempa\caption@tempb
1327           \expandafter\caption@if@empty\expandafter{\@gobbletwo#1}%
1328         \else
1329           \def\caption@tempb{\index}%
1330           \ifx\caption@tempa\caption@tempb
1331             \expandafter\caption@if@empty\expandafter{\@gobbletwo#1}%
1332           \else
1333             \def\caption@tempb{\glossary}%
1334             \ifx\caption@tempa\caption@tempb
1335               \expandafter\caption@if@empty\expandafter{\@gobbletwo#1}%
1336             \else
1337               \let\caption@ifempty\@gobbletwo
1338             \fi
1339           \fi
1340         \fi
1341       \fi
1342     \fi
1343   \fi}

1344 \long\def\caption@car#1#2\caption@nil{#1}% same as \@car, but \long

```

\caption@@par \caption@@par

This command will be executed with every \par inside the caption.

```

1345 \newcommand*\caption@@par{%
1346   \parindent\caption@parindent\hangindent\caption@hangindent}%

```

## 23 Types & sub-types

```

\DeclareCaptionType \DeclareCaptionType[<options>]{<environment>}[<name>][<list name>]
1347 \newcommand*\DeclareCaptionType{%

```



```

1348 \@testopt\@DeclareCaptionType{}
1349 \@onlypreamble\DeclareCaptionType

1350 \def\@DeclareCaptionType[#1]#2{%
1351   \def\caption@type{#2}%
1352   \caption@Debug{New type `#2'}%
1353   \newcounter{#2}\@namedef{theH#2}{}%
1354   \KV@caption@DCT@within\caption@within@default
1355   \caption@DeclareWithinOption{#2}%
1356   \KV@caption@DCT@placement{tbp}%
1357   \caption@ifundefined\c@float@type
1358     {\newcounter{float@type}%
1359      \setcounter{float@type}{\caption@ifundefined\c@figure14}}%
1360   {}%
1361   \caption@Debug{float type `#2'=\the\value{float@type}}%
1362   \expandafter\xdef\csname ftype#2\endcsname{\the\value{float@type}}%
1363   \addtocounter{float@type}{\value{float@type}}%
1364   \KV@caption@DCT@fileext{lo#2}%
1365   \@namedef{fnum#2}{\@nameuse{#2name}\nobreakspace\@nameuse{the#2}}%
1366   \newenvironment{#2}{\@float{#2}}{\end@float}%
1367   \newenvironment{#2*}{\@dblfloat{#2}}{\end@dblfloat}%
1368   \expandafter\newcommand\csname listof#2s\endcsname{\caption@listof{#2}}%
1369   \expandafter\newcommand\csname listof#2es\endcsname{\caption@listof{#2}}%
1370   \caption@ifundefined\l@figure
1371     {\@namedef{l@#2}{\@dottedtocline{1}{1.5em}{2.3em}}}%
1372     {\expandafter\let\csname l@#2\endcsname\l@figure}%

1373   \expandafter\newcommand\csname #2name\endcsname{}%
1374   \edef\@tempa{\def\noexpand\@tempa{\@car#2\@nil}}%
1375   \uppercase\expandafter{\@tempa}%
1376   \edef\@tempb{\noexpand\g@addto@macro\noexpand\@tempa{\@cdr#2\@nil}}%
1377   \@tempb
1378   \expandafter\let\csname #2name\endcsname\@tempa
1379   \expandafter\newcommand\csname list#2name\endcsname{}%
1380   \expandafter\xdef\csname list#2name\endcsname{List of \@tempa s}%

1381   \@cons\caption@typelist{#2}%
1382   \caption@setkeys[caption]{caption@DCT}{#1}%

1383   \caption@ifundefined\float@exts{\newtoks\float@exts}{}%
1384   \let\float@do=\relax
1385   \edef\@tempa{\noexpand\float@exts{\the\float@exts\float@do{\@nameuse{ext#2}}}}%
1386   \@tempa
1387   \caption@ifundefined\float@addtolists{%
1388     \newcommand\float@addtolists[1]{%
1389       \def\float@do###1{\addtocontents{###1}{##1}}\the\float@exts}%
1390     \caption@ifundefined\@chapter{\caption@PatchChapter}}}%

1391   \@ifnextchar[\@@DeclareCaptionType\relax}
1392 \@onlypreamble\@DeclareCaptionType

1393 \def\@@DeclareCaptionType[#1]{%
1394   \KV@caption@DCT@name{#1}%
1395   \@ifnextchar[\@@@DeclareCaptionType\relax}
1396 \@onlypreamble\@@DeclareCaptionType

1397 \def\@@@DeclareCaptionType[#1]{%
1398   \KV@caption@DCT@listname{#1}}
1399 \@onlypreamble\@@@DeclareCaptionType

```

```

1400 \let\DeclareFloatingEnvironment\DeclareCaptionType % alternative command name
1401 \@onlypreamble\DeclareFloatingEnvironment

\caption@within@default The default ‘within’ value.
1402 \newcommand*\caption@within@default{\caption@ifundefined\c@chapter{none}{chapter}}
1403 \@onlypreamble\caption@within@default

\caption@listof \caption@listof{<float type>}
1404 \newcommand*\caption@listof[1]{%
1405   \begingroup
1406     \expandafter\let\expandafter\listfigurename\csname list#1name\endcsname
1407     \expandafter\let\expandafter\ext@figure\csname ext@#1\endcsname
1408     \let\caption@ORI@starttoc\@starttoc
1409     \renewcommand*\@starttoc[1]{%
1410       \expandafter\caption@ORI@starttoc\expandafter{\ext@figure}}%
1411     \listoffigures
1412   \endgroup}

\caption@typelist An \@elt-list containing the caption types defined with \DeclareCaptionType.
1413 \newcommand*\caption@typelist{}

The available <options> are: fileext=<file extension>, listname=<list name>, name=<prosa
name>, placement=<htbp>, within=<none,chapter,section>, and without.

1414 \define@key{caption@DCT}{fileext}{\@namedef{ext@\caption@type}{#1}}
1415 \@onlypreamble@key{caption@DCT}{fileext}
1416 \define@key{caption@DCT}{listname}{\@namedef{list\caption@type name}{#1}}
1417 \@onlypreamble@key{caption@DCT}{listname}
1418 \define@key{caption@DCT}{name}{\@namedef{\caption@type name}{#1}}
1419 \@onlypreamble@key{caption@DCT}{name}
1420 \define@key{caption@DCT}{placement}{\@namedef{fps@\caption@type}{#1}}
1421 \@onlypreamble@key{caption@DCT}{placement}
1422 \define@key{caption@DCT}{within}{%
1423   \caption@ifundefined\c@chapter{\@removefromreset\caption@type{chapter}}%
1424   \@removefromreset\caption@type{section}%
1425   \begingroup
1426     \caption@setkeys[caption]{caption@within}{#1}%
1427   \endgroup}
1428 %\@onlypreamble@key{caption@DCT}{within}
1429 \define@key{caption@DCT}{without}{\KV@caption@DCT@within{none}}
1430 %\@onlypreamble@key{caption@DCT}{without}

1431 \define@key{caption@within}{none}[]{%
1432   \caption@within{}}{}
1433 %\@onlypreamble@key{caption@within}{none}
1434 \define@key{caption@within}{section}[]{%
1435   \@addtoreset\caption@type{section}%
1436   \caption@within{\ifnum\c@section>\z@ \thesection.\fi}{\theHsection.}}
1437 %\@onlypreamble@key{caption@within}{section}
1438 \caption@ifundefined\c@chapter{\%
1439   \define@key{caption@within}{chapter}[]{%
1440     \@addtoreset\caption@type{chapter}%
1441     \caption@within{\ifnum\c@chapter>\z@ \thechapter.\fi}{\theHchapter.}}
1442 }% \@onlypreamble@key{caption@within}{chapter}}

```

```

\caption@within \caption@within{\theHcode}{\theHcode}
1443 \newcommand*\caption@within{%
1444 \expandafter\caption@within@\expandafter{\caption@type}}
1445 %\@onlypreamble\caption@within
1446 \newcommand*\caption@within@[3]{%
1447 \global\@namedef{the#1}{#2\arabic{#1}}%
1448 \@ifundefined{theH#1}\caption@AtBeginDocument\@firstofone
1449 {\global\@namedef{theH#1}{#3\arabic{#1}}}%
1450 %\@onlypreamble\caption@within@

\@removefromreset This code was taken from the remreset package which is part of the ‘carlisle’ package
bundle. (Copyright 1997 David Carlisle)
1451 \providecommand*\@removefromreset[2]{%
1452 \expandafter\let\csname c@#1\endcsname\@removefromreset
1453 \def\@elt##1{%
1454 \expandafter\ifx\csname c@##1\endcsname\@removefromreset
1455 \else
1456 \noexpand\@elt{##1}%
1457 \fi}%
1458 \expandafter\xdef\csname cl@#2\endcsname{%
1459 \csname cl@#2\endcsname}}

\caption@PatchChapter We try to patch \chapter so \float@addtolists will be supported. (Note: The
KOMA-Script classes already support \float@addtolists.)
1460 \newcommand*\caption@PatchChapter{%
1461 \providecommand*\@chapterlistsgap{10\p@}%
1462 % report.cls [2005/09/16 v1.4f Standard LaTeX document class]
1463 \caption@patch@chapter{report}{%
1464 \ifnum \c@secnumdepth >\m@ne
1465 \refstepcounter{chapter}%
1466 \typeout{\@chapapp\space\thechapter.}%
1467 \addcontentsline{toc}{chapter}%
1468 {\protect\numberline{\thechapter}##1}%
1469 \else
1470 \addcontentsline{toc}{chapter}{##1}%
1471 \fi
1472 \chaptermark{##1}%
1473 \addtocontents{lof}{\protect\addvspace{10\p@}}%
1474 \addtocontents{lot}{\protect\addvspace{10\p@}}%
1475 \if@twocolumn
1476 \topnewpage[\@makechapterhead{##2}]%
1477 \else
1478 \makechapterhead{##2}%
1479 \afterheading
1480 \fi
1481 }{%
1482 \ifnum \c@secnumdepth >\m@ne
1483 \refstepcounter{chapter}%
1484 \typeout{\@chapapp\space\thechapter.}%
1485 \addcontentsline{toc}{chapter}%
1486 {\protect\numberline{\thechapter}##1}%
1487 \else
1488 \addcontentsline{toc}{chapter}{##1}%

```

```

1489 \fi
1490 \chaptermark{##1}%
1491 \ifdim \@chapterlistsgap>\z@
1492 \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1493 \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1494 \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1495 \fi
1496 \if@twocolumn
1497 \topnewpage[\@makechapterhead{##2}]%
1498 \else
1499 \makechapterhead{##2}%
1500 \afterheading
1501 \fi}%

1502 % book.cls [2005/09/16 v1.4f Standard LaTeX document class]
1503 \caption@patch@chapter{book}{%
1504 \ifnum \c@secnumdepth >\m@ne
1505 \if@mainmatter
1506 \refstepcounter{chapter}%
1507 \typeout{\@chapapp\space\thechapter.}%
1508 \addcontentsline{toc}{chapter}%
1509 {\protect\numberline{\thechapter}##1}%
1510 \else
1511 \addcontentsline{toc}{chapter}{##1}%
1512 \fi
1513 \else
1514 \addcontentsline{toc}{chapter}{##1}%
1515 \fi
1516 \chaptermark{##1}%
1517 \addtocontents{lof}{\protect\addvspace{10\p@}}%
1518 \addtocontents{lot}{\protect\addvspace{10\p@}}%
1519 \if@twocolumn
1520 \topnewpage[\@makechapterhead{##2}]%
1521 \else
1522 \makechapterhead{##2}%
1523 \afterheading
1524 \fi
1525 }{%
1526 \ifnum \c@secnumdepth >\m@ne
1527 \if@mainmatter
1528 \refstepcounter{chapter}%
1529 \typeout{\@chapapp\space\thechapter.}%
1530 \addcontentsline{toc}{chapter}%
1531 {\protect\numberline{\thechapter}##1}%
1532 \else
1533 \addcontentsline{toc}{chapter}{##1}%
1534 \fi
1535 \else
1536 \addcontentsline{toc}{chapter}{##1}%
1537 \fi
1538 \chaptermark{##1}%
1539 \ifdim \@chapterlistsgap>\z@
1540 \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1541 \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1542 \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%

```

```

1543 \fi
1544 \if@twocolumn
1545 \topnewpage[\@makechapterhead{##2}]%
1546 \else
1547 \makechapterhead{##2}%
1548 \afterheading
1549 \fi}%

1550 % amsbook.cls [2004/08/06 v2.20]
1551 % smfbook.cls [1999/11/15 v1.2f Classe LaTeX pour les monographies editees par
1552 \caption@patch@chapter{ams/smfbook}{%
1553 \refstepcounter{chapter}%
1554 \ifnum\c@secnumdepth<\z@ \let\@secnumber\@empty
1555 \else \let\@secnumber\thechapter \fi
1556 \typeout{\chaptername\space\@secnumber}%
1557 \def\@toclevel{0}%
1558 \ifx\chaptername\appendixname \@tocwriteb\tocappendix{chapter}{##2}%
1559 \else \@tocwriteb\tocchapter{chapter}{##2}\fi
1560 \chaptermark{##1}%
1561 \addtocontents{lof}{\protect\addvspace{10\p@}}%
1562 \addtocontents{lot}{\protect\addvspace{10\p@}}%
1563 \makechapterhead{##2}\@afterheading
1564 }{%
1565 \refstepcounter{chapter}%
1566 \ifnum\c@secnumdepth<\z@ \let\@secnumber\@empty
1567 \else \let\@secnumber\thechapter \fi
1568 \typeout{\chaptername\space\@secnumber}%
1569 \def\@toclevel{0}%
1570 \ifx\chaptername\appendixname \@tocwriteb\tocappendix{chapter}{##2}%
1571 \else \@tocwriteb\tocchapter{chapter}{##2}\fi
1572 \chaptermark{##1}%
1573 \ifdim \@chapterlistsgap>\z@
1574 \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1575 \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1576 \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1577 \fi
1578 \makechapterhead{##2}\@afterheading}%

1579 % scrreprt/scrbook.cls
1580 \caption@ifundefined\KOMAClassName{}{%
1581 \caption@Debug{document class '\KOMAClassName' detected}%
1582 \let\caption@patch@chapter\@gobblethree}%

1583 % rapport1/3.cls [2004/06/07 v2.1a NTG LaTeX document class]
1584 \caption@patch@chapter{rapport}{%
1585 \ifnum \c@secnumdepth >\m@ne
1586 \refstepcounter{chapter}%
1587 \typeout{\@chapapp\space\thechapter.}%
1588 \addcontentsline{toc}{chapter}%
1589 {\protect\numberline{\thechapter}\toc@font0 ##1}%
1590 \else
1591 \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1592 \fi
1593 \chaptermark{##1}%
1594 \addtocontents{lof}{\protect\addvspace{10\p@}}%
1595 \addtocontents{lot}{\protect\addvspace{10\p@}}%

```

```

1596 \if@twocolumn
1597 \topnewpage[\@makechapterhead{##2}]%
1598 \else
1599 \@makechapterhead{##2}%
1600 \afterheading
1601 \fi
1602 }{%
1603 \ifnum \c@secnumdepth >\m@ne
1604 \refstepcounter{chapter}%
1605 \typeout{\@chapapp\space\thechapter.}%
1606 \addcontentsline{toc}{chapter}%
1607 {\protect\numberline{\thechapter}\toc@font0 ##1}%
1608 \else
1609 \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1610 \fi
1611 \chaptermark{##1}%
1612 \ifdim \@chapterlistsgap>\z@
1613 \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1614 \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1615 \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1616 \fi
1617 \if@twocolumn
1618 \topnewpage[\@makechapterhead{##2}]%
1619 \else
1620 \@makechapterhead{##2}%
1621 \afterheading
1622 \fi}%

1623 % boek(3).cls [2004/06/07 v2.1a NTG LaTeX document class]
1624 \caption@patch@chapter{boek}{%
1625 \ifnum \c@secnumdepth >\m@ne
1626 \if@mainmatter
1627 \refstepcounter{chapter}%
1628 \typeout{\@chapapp\space\thechapter.}%
1629 \addcontentsline{toc}{chapter}%
1630 {\protect\numberline{\thechapter}\toc@font0 ##1}%
1631 \else
1632 \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1633 \fi
1634 \else
1635 \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1636 \fi
1637 \chaptermark{##1}%
1638 \addtocontents{lof}{\protect\addvspace{10\p@}}%
1639 \addtocontents{lot}{\protect\addvspace{10\p@}}%
1640 \if@twocolumn
1641 \topnewpage[\@makechapterhead{##2}]%
1642 \else
1643 \@makechapterhead{##2}%
1644 \afterheading
1645 \fi
1646 }{%
1647 \ifnum \c@secnumdepth >\m@ne
1648 \if@mainmatter
1649 \refstepcounter{chapter}%

```

```

1650         \typeout{\@chapapp\space\thechapter.}%
1651         \addcontentsline{toc}{chapter}%
1652             {\protect\numberline{\thechapter}\toc@font0 ##1}%
1653     \else
1654         \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1655     \fi
1656 \else
1657     \addcontentsline{toc}{chapter}{\toc@font0 ##1}%
1658 \fi
1659 \chaptermark{##1}%
1660 \ifdim \@chapterlistsgap>\z@
1661     \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1662     \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1663     \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1664 \fi
1665 \if@twocolumn
1666     \topnewpage[\@makechapterhead{##2}]%
1667 \else
1668     \@makechapterhead{##2}%
1669     \@afterheading
1670 \fi}%

1671 % thesis.cls [1996/25/01 1.0g LaTeX document class (wm).]
1672 \caption@patch@chapter{thesis}{%
1673     \ifnum \c@secnumdepth >\m@ne
1674         \if@mainmatter
1675             \refstepcounter{chapter}%
1676             \typeout{\chaptername\space\thechapter.}
1677             \if@thema
1678                 \ifx\@shortauthor\@empty
1679                     \addcontentsline{toc}{chapter}{%
1680                         \protect\numberline{\thechapter.}##1}%
1681                 \else
1682                     \addcontentsline{toc}{chapter}{%
1683                         \protect\numberline{\thechapter.}%
1684                         \@shortauthor\hfill\mbox{}\vskip\normallineskip ##1}%
1685                 \fi
1686             \else
1687                 \addcontentsline{toc}{chapter}{%
1688                     \protect\numberline{\thechapter.}##1}%
1689             \fi
1690         \else
1691             \addcontentsline{toc}{chapter}{##1}
1692         \fi
1693     \else
1694         \addcontentsline{toc}{chapter}{##1}
1695     \fi
1696     \chaptermark{##1}
1697     \addtocontents{lof}{\protect\addvspace{10pt}}
1698     \addtocontents{lot}{\protect\addvspace{10pt}}
1699     \if@twocolumn
1700         \topnewpage[\@makechapterhead{##2}]
1701     \else
1702         \@makechapterhead{##2}
1703         \@afterheading

```

```

1704 \fi
1705 }{%
1706 \ifnum \c@secnumdepth >\m@ne
1707 \if@mainmatter
1708 \refstepcounter{chapter}%
1709 \typeout{\chaptername\space\thechapter.}%
1710 \if@thema
1711 \ifx\@shortauthor\@empty
1712 \addcontentsline{toc}{chapter}{%
1713 \protect\numberline{\thechapter.}##1}%
1714 \else
1715 \addcontentsline{toc}{chapter}{%
1716 \protect\numberline{\thechapter.}%
1717 \@shortauthor\hfill\mbox{}}\vskip\normallineskip ##1}%
1718 \fi
1719 \else
1720 \addcontentsline{toc}{chapter}{%
1721 \protect\numberline{\thechapter.}##1}%
1722 \fi
1723 \else
1724 \addcontentsline{toc}{chapter}{##1}%
1725 \fi
1726 \else
1727 \addcontentsline{toc}{chapter}{##1}%
1728 \fi
1729 \chaptermark{##1}%
1730 \ifdim \@chapterlistsgap>\z@
1731 \addtocontents{lof}{\protect\addvspace{\@chapterlistsgap}}%
1732 \addtocontents{lot}{\protect\addvspace{\@chapterlistsgap}}%
1733 \float@addtolists{\protect\addvspace{\@chapterlistsgap}}%
1734 \fi
1735 \if@twocolumn
1736 \@topnewpage[\@makechapterhead{##2}]%
1737 \else
1738 \@makechapterhead{##2}%
1739 \@afterheading
1740 \fi}%
1741 \ifx\caption@patch@chapter\@gobblethree \else
1742 \caption@Debug{%
1743 Unsupported document class detected,\MessageBreak
1744 or \noexpand\@chapter was redefined by another package}%
1745 \fi
1746 \let\caption@PatchChapter\@undefined}
1747 %\@onlypreamble\caption@PatchChapter
1748 \newcommand\caption@patch@chapter[3]{%
1749 \begingroup
1750 % \let\if@twocolumn\iffalse
1751 \let\if@mainmatter\iffalse
1752 \let\if@thema\iffalse
1753 \def\@tempa[##1]##2{##2}%
1754 \ifx\@tempa\@chapter
1755 \caption@Debug{document class `#1' detected}%
1756 \gdef\@chapter[##1]##2{##3}%

```



```

1757     \global\let\caption@patch@chapter\@gobblethree
1758     \fi
1759     \endgroup}
1760 %\@onlypreamble\caption@patch@chapter
1761 \long\def \@gobblethree #1#2#3{}

```

\@stpelt We patch \@stpelt so a list of ‘connected’ counters will be reset, too. (Like \stepcounter does in ltcoun.ts.dtx.)

```

1762 \newcommand*\caption@patch@stpelt{%
1763   \let\caption@stpelt\@stpelt
1764   \def\@stpelt##1{%
1765     \caption@stpelt{##1}%
1766     \begingroup
1767       \let\@elt\caption@stpelt
1768       \csname caption@cl@##1\endcsname
1769     \endgroup}%
1770   \let\caption@patch@stpelt\relax}
1771 \@onlypreamble\caption@patch@stpelt

```

\caption@addtoreset Like \@addtoreset from ltcoun.ts.dtx

```

1772 \newcommand*\caption@addtoreset[2]{%
1773   \caption@patch@stpelt
1774   \@ifundefined{caption@cl@#2}{\@namedef{caption@cl@#2}{}{}}{}%
1775   \expandafter\@cons\csname caption@cl@#2\endcsname{{#1}}
1776 \@onlypreamble\caption@addtoreset

```

\caption@addtoreset Like \@removefromreset from remreset.sty

```

1777 \newcommand*\caption@removefromreset[2]{%
1778   \begingroup
1779     \expandafter\let\csname c@#1\endcsname\caption@removefromreset
1780     \def\@elt##1{%
1781       \expandafter\ifx\csname c@##1\endcsname\caption@removefromreset
1782       \else
1783         \noexpand\@elt{##1}%
1784       \fi}%
1785     \expandafter\xdef\csname caption@cl@#2\endcsname{%
1786       \csname caption@cl@#2\endcsname}%
1787   \endgroup}
1788 \@onlypreamble\caption@removefromreset

```

\DeclareCaptionSubType \DeclareCaptionSubType[*numbering scheme*]{*type*}  
\DeclareCaptionSubType\*[*numbering scheme*]{*type*}

The starred variant provides the numbering format *type*.*subtype* while the non-starred variant simply uses *subtype*.

```

1789 \newcommand*\DeclareCaptionSubType{%
1790   \caption@teststar\caption@declaresubtype\@firstoftwo\@secondoftwo}
1791 \@onlypreamble\DeclareCaptionSubType
1792 \newcommand*\caption@declaresubtype[1]{%
1793   \@testopt{\caption@@declaresubtype{#1}}{alph}}
1794 \@onlypreamble\caption@declaresubtype
1795 \def\caption@@declaresubtype#1[#2]#3{%
1796   \@ifundefined{c@#3}%
1797   {\caption@Error{No float type ‘#3’ defined}}%

```

```

1798     {\ifundefined{c@sub#3}%
1799       {\caption@Debug{New subtype `sub#3'}%
1800         \newcounter{sub#3}%
1801         \caption@addtoreset{sub#3}{#3}%
1802         \@namedef{ext@sub#3}{\csname ext@#3\endcsname}%
1803         \caption@declaresublistentry{#3}%
1804         \@cons\caption@subtypelist{{#3}}}%
1805       {\caption@Debug{Modify caption `sub#3'}}}%

1806     \@namedef{sub#3name}{}%
1807     \@namedef{sub#3autorefname}{\csname #3name\endcsname}%
1808     #1% is \@firstoftwo in star form, and \@secondoftwo otherwise
1809     {\@namedef{p@sub#3}{}%
1810       \@namedef{thesub#3}{\csname the#3\endcsname.\@nameuse{#2}{sub#3}}}%
1811     {\@namedef{p@sub#3}{\csname the#3\endcsname}%
1812       \@namedef{thesub#3}{\@nameuse{#2}{sub#3}}}%
1813     \@namedef{theHsub#3}{\csname theH#3\endcsname.\arabic{sub#3}}%
1814     }}

1815 \@onlypreamble\caption@@declaresubtype

1816 \newcommand*\caption@declaresublistentry{%
1817   \caption@ifundefined\l@chapter
1818     {\caption@@declaresublistentry\l@subsubsection}%
1819     {\caption@@declaresublistentry\l@section}}
1820 \@onlypreamble\caption@declaresublistentry

1821 \newcommand*\caption@@declaresublistentry[2]{%
1822   \ifx#1\@undefined
1823     \caption@@@declaresublistentry\relax\@dottedtocline\caption@nil{#2}%
1824   \else
1825     \expandafter\caption@@@declaresublistentry#1{}{}\@dottedtocline\caption@nil{#
1826   \fi}
1827 \@onlypreamble\caption@@declaresublistentry

1828 \long\def\caption@@@declaresublistentry#1\@dottedtocline#2\caption@nil#3{%
1829   \def\@tempa{#1}%
1830 % Does \l@ (sub)subsection start with \@dottedtocline?
1831   \ifx\@tempa\@empty
1832 % Yes
1833     \caption@@@declaresublistentry{#3}#2\caption@nil
1834   \else
1835 % No
1836     \caption@@@declaresublistentry{#3}@\{3.8em\}{3.2em}\caption@nil
1837   \fi}
1838 \@onlypreamble\caption@@@declaresublistentry

1839 \def\caption@@@declaresublistentry#1#2#3#4#5\caption@nil{%
1840   \expandafter\caption@@@@declaresublistentry\expandafter
1841     {\csname @dotted\csname ext@#1\endcsname line\endcsname}{#1}{#3}{#4}}
1842 \@onlypreamble\caption@@@@declaresublistentry

1843 \newcommand*\caption@@@@declaresublistentry[4]{%
1844   \@namedef{l@sub#2}{#1{2}{#3}{#4}}%
1845   \caption@@@@@declaresublistentry#1{c\csname ext@#2\endcsname depth}%
1846 \@onlypreamble\caption@@@@@declaresublistentry

1847 \newcommand*\caption@@@@@declaresublistentry[2]{
1848   \ifx#1\relax

```

```

1849 \def#1#1{%
1850 \def\next{\@dottedtocline{##1}}%
1851 \@ifundefined{#2}{}{%
1852 \ifnum #1>\@nameuse{#2}\relax
1853 \let\next\@gobblefour
1854 \fi}%
1855 \next}%
1856 \fi}
1857 \@onlypreamble\caption@@@@declaresublistentry

\caption@subtypelist An \@elt-list containing the subtypes defined with \DeclareCaptionSubType.
1858 \newcommand*\caption@subtypelist{}

\caption@For \caption@For{<elt-list>}{<code with #1>}
\caption@For*{<elt-list>}{<code with #1>}
1859 \newcommand*\caption@For{\caption@withoptargs\caption@For}
1860 %\@onlypreamble\caption@For

1861 \newcommand\caption@@For[3]{%
1862 \caption@AtBeginDocument#1{%
1863 \def\@elt##1{#3}%
1864 \@nameuse{caption@#2}%
1865 \let\@elt\relax}}%
1866 %\@onlypreamble\caption@@For

```

## 24 subfig package adoptions

We have to make several adoptions to the caption kernel *v1.1* here.

```

1867 \caption@AtBeginDocument{%
1868 \def\@tempa{\@ifstar\sf@@subref\sf@subref}%
1869 \ifx\subref\@tempa
1870 \caption@InfoNoLine{subfig package 1.2 or 1.3 is loaded}%
1871 \let\caption@setfloattype\@gobble
1872 \let\@dottedxxxline\sf@NEW@dottedxxxline
1873 \let\sf@subfloat\sf@NEW@subfloat
1874 \fi
1875 \let\sf@NEW@dottedxxxline\@undefined
1876 \let\sf@NEW@subfloat\@undefined}
1877 \def\sf@NEW@dottedxxxline#1#2#3#4#5#6#7{%
1878 \begingroup
1879 \caption@setfloattype{#1}%
1880 \caption@setoptions{subfloat}%
1881 \caption@setoptions{sub#1}%
1882 \ifnum #3>\@nameuse{c@#2depth}\else
1883 \dottedtocline{\z@}{#4}{#5}{#6}{#7}%
1884 \fi
1885 \endgroup}
1886 \def\sf@NEW@subfloat{%
1887 \begingroup
1888 \caption@setfloattype\@capttype
1889 \sf@ifpositiontop{%

```

```

1890     \maincaptiontoptrue
1891   }{%
1892     \maincaptiontopfalse
1893   }%
1894   \caption@setoptions{subfloat}%
1895   \caption@setoptions{sub\@capttype}%
1896   \let\sf@oldlabel=\label
1897   \let\label=\subfloat@label
1898   \ifmaincaptiontop\else
1899     \advance\@nameuse{c@\@capttype}\@ne
1900   \fi
1901   \refstepcounter{sub\@capttype}%
1902   \setcounter{sub\@capttype @save}{\value{sub\@capttype}}%
1903   \@ifnextchar [% %] match left bracket
1904     {\sf@@subfloat}%
1905     {\sf@@subfloat[\@empty]}

```

## References

- [1] Anne Brüggemann-Klein:  
*Einführung in die Dokumentverarbeitung*,  
B.G. Teubner, Stuttgart, 1989
- [2] Sebastian Rahtz & Heiko Oberdiek:  
*Hypertext marks in L<sup>A</sup>T<sub>E</sub>X*,  
November 12, 2007
- [3] Heiko Oberdiek:  
*The refcount package*,  
2006/02/20
- [4] Donald Arseneau:  
*Three part tables: title, tabular environment, notes*,  
2003/06/13